Performance Analysis of a Lightweight NEMO Implementation for Low-End devices

J.L.Almodovar[*]
Dpto. Ing. Telemática
Univ. Carlos III de Madrid
Madrid, Spain
jalmodov@it.uc3m.es

A. de la Oliva
Dpto. Ing. Telemática
Univ. Carlos III de Madrid
Madrid, Spain
aoliva@it.uc3m.es

C. Lazo Ram
Inst. de Informática
Univ. Austral de Chile
Chile
clazo@inf.uach.cl

C. Cervelló-Pastor[†]
Dpto. Ing. Telemática
Univ. Politécnica de Catalunya
Spain
cristina@entel.upc.edu

ABSTRACT

IETF Network Mobility (NEMO) Basic Support Protocol is an IP mobility management protocol designed to provide seamless IP mobility to complete networks. This protocol is of crucial interest in vehicular mobility and it is being applied in several research projects and initiatives which pretend to incorporate communication support to the transportation system. One of the problems which appear when including NEMO on vehicles is the need of incorporate network infrastructure to the vehicle with its associated cost in terms of money, space etc. On this work we present an implementation of the NEMO protocol specifically designed to run on low end network devices. We also present the performance figures of it running on a reference device, proving that its performance is enough to comply the requirements of vehicular communications.

Categories and Subject Descriptors

D.m [**Software**]: Miscellaneous

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Network Mobility, low-end devices, performance analysis

1. INTRODUCTION

In the recent past the Internet scenario has changed critically. From fixed wired networks which were the majority of the deployed networks some years ago, now Internet is formed by a huge range of wired and wireless technologies. This fact joined to the recent appearance of multi-interface mobile terminals which apart of being able to use cellular technologies such as GSM, GPRS or UMTS are able to use IP based networks such as WiFi, moved the research trends to the mobility topic. Current standard Internet protocols, such as IPv4, do no support transparent mobility, since IP was not designed taking into account mobility. Terminals were considered to be fixed, and the IP address plays the role of both identifier and locator in a network, so a change of the address (needed when connecting to a different subnet) implies a change of the identifier which breaks ongoing transport connections. Protocols such as DHCP [1] enable portability (i.e., a terminal can change its point of attachment and obtain connectivity, but all its connections should be restarted) but this solution does not maintain open connections or allows the node to be reachable while changing its point of attachment. By mobility, we mean enabling the transparent movement of nodes, without breaking ongoing connections and allowing the nodes to be reachable through a permanent IP address. IP mobility has been a hot research topic during past years. Several working groups of the IETF[1] have studied the mobility problem from several perspectives. One of the most successful protocols on the mobility area has been Mobile IPv6[8], designed by the mip6 IETF WG and now inherited by the new mext WG[2]. This protocol enables the client terminals to support basic IPv6 mobility while maintaining ongoing sessions and global

---

[*] J.L Almodovar and A. de la Oliva's work was partially funded by the European Project DAIDALOS II. DAIDALOS II receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability

[†] C. Lazo Ram and C. Cervell's work was partially supported by the Spanish Government under the POSEIDON (TSI2006-12507-C03-01) project

[1] www.itef.org

[2] www.ietf.org/html.charters/mext-charter.html

reachability of the terminal. The concepts acquired during the conception of Mobile IPv6 were used to design an IP Network Mobility solution called NEMO (Network Mobility)[4]. This protocol allows complete networks to change their point of attachment to the Internet, with the benefit of not requiring any change on the nodes attached to the network that moves. NEMO relies on two entities, the Mobile Router and the Home Agent. The Mobile Router is the node in charge of managing the mobility of the network and it acts as the gateway to Internet. The Home Agent is a node deployed on the Home Network (the network to which the original point of attachment to Internet of the Mobile Network belongs to) which knows the current point of attachment of the Mobile Network and forwards packets destined to it to the new point of attachment. Network Mobility has opened new paradigms of applicability, network paradigms such as Wireless Personal Area Network (WPAN), Wireless Body Networks (WBAN) or Wireless Vehicular Networks (WVAN) can benefit from the application of this protocol. One of the areas were NEMO has been particularly interesting is Vehicular networks. Under this application paradigm, several research projects consider NEMO or NEMO variants as the main protocol to provide vehicular communications. One of the current limitations for a wide adoption of vehicular communication solutions is the added cost of deploying networking hardware to the vehicle. Current reference implementations of the Mobile Router are focused on computers, which are not suitable to deploy on a car due to price, size and power constrains. These limitations could be saved if NEMO is deployed on a small, low power consuming and low-cost device. The aim of this work is to present a NEMO implementation specifically designed to run on low-cost devices suitable for mass market production. Several performance measures are provided, proving that this kind of devices, along with the NEMO implementation, are powerful enough to enable Network Mobility of the devices attached to the vehicle, and at a low price suitable for all needs.
The reminder of this paper is organized as follows. Section 2 presents a description of the NEMO protocol and the current projects which are particularly interested on its deploy. Section 3 explains the characteristics and objectives of the implementation, along with the reference device selected. Section 4 details the implementation and its limitations. Section 5 shows the performance results making a comparison the implementation performance with the obtained using traditional computers. Finally section 6 concludes this work.

2. NETWORK MOBILITY BASIC SUPPORT PROTOCOL

Providing mobility at IP-level is difficult because IP addresses play the role of identifier and locator. Routing in IP is hierarchical, and IP addresses are configured taking into account the network that the nodes are attached to. Routers in a network forward packets based on the destination address and the information stored in their routing tables. When a node changes its point of attachment, packets addressed to that node are delivered (using normal IP routing) to the network it was connected to. In order to be able to receive packets at its new location, the node should configure an IP address belonging to the address space of the new network, but this implies changing also the addresses that transport protocols use (IP addresses are part of transport addresses), which breaks established sessions. There are some situations in which not only a single node moves, but a complete network does. This will become more and more usual as the demand for ubiquitous Internet access in public transportation systems increases. In more precise terms, a Network that Moves (NEMO) - a mobile network - can be defined as a network whose attachment point to the Internet varies with time. The router within the NEMO that connects to the Internet is called the Mobile Router (MR). It is assumed that the NEMO has a Home Network to which the permanent address associated with the MR belongs to. Since the NEMO is reachable through the Home Network, the Mobile Network has configured addresses belonging to an address block assigned to the Home Network. These addresses remain assigned to the NEMO when it is away from home. Naturally, these addresses only have topological meaning when the NEMO is at home. When the NEMO is away from home, packets addressed to the Mobile Network Nodes (MNNs) will still be routed to the Home Network. Additionally, when the NEMO is away from home, i.e., it is in a visited network, the MR acquires an address from the visited network, called the Care-of Address (CoA), where the routing architecture can deliver packets without additional mechanisms. The goal of the network mobility support mechanisms is to preserve established communications between the MNNs and external Correspondent Nodes (CNs) through movement. Packets of such communications will be addressed to the MNNs addresses, which belong to the Mobile Network Prefix (MNP), so additional mechanisms to forward packets between the Home Network and the NEMO are needed. The basic solution for network mobility support [4] essentially creates a bi-directional tunnel between a special node located in the Home Network of the NEMO (the Home Agent, HA), and the Care-of Address of the MR (figure 1).

This basic solution is derived from the solution proposed for host mobility support, MIPv6 [8], without including the Route Optimization support. Actually, the protocol is similar and the mobility signaling (i.e., Binding Update (BU) message) is extended to inform the Home Agent about the IP address of the NEMO side of the tunnel (that is, the CoA of the MR), through which the HA has to forward the packets addressed to the Mobile Network Prefix. In addition to the triangular routing problem (all packets pass through the HA), also present in Mobile IPv6, the NEMO Basic Support protocol introduces the so-called *pinball* routing, that appears when nesting is considered. A Mobile Network can be attached to another Mobile Network, thus forming nested chains of networks (figure 2).

2.1 Importance of NEMO in vehicular mobility research

The importance of providing a light weight NEMO implementation for low-end devices can be understood by analyzing the current research projects on vehicular communications which use NEMO or use ideas inherited from this protocol on its architectures. On the following lines, an overview of several projects using NEMO is provided.

- One of the usage models of NEMO is to provide Intel-

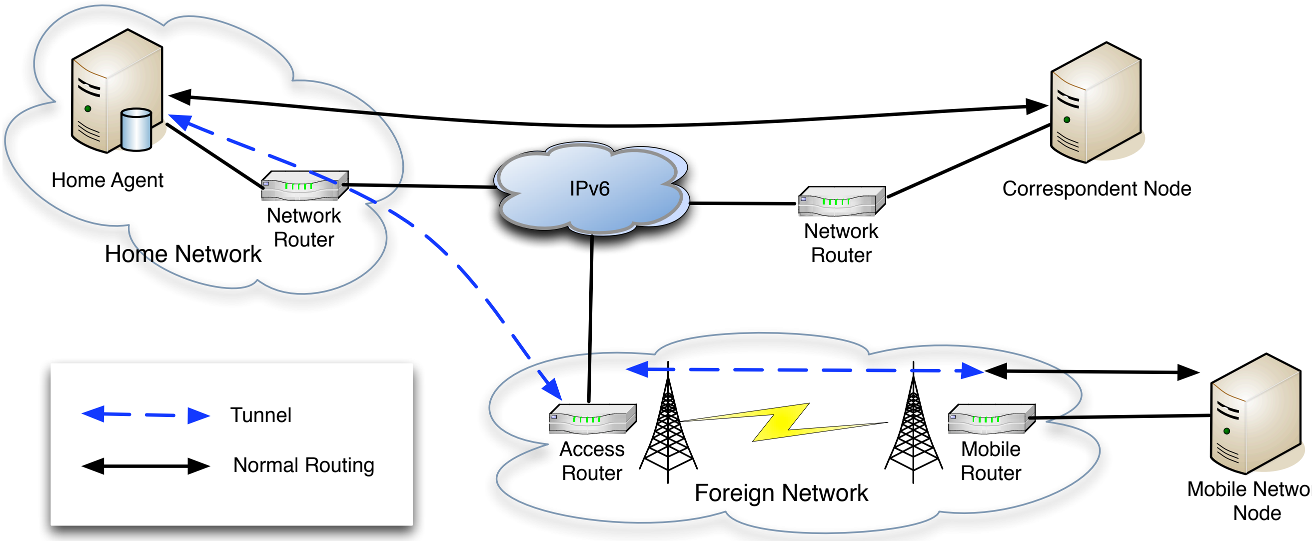

Figure 1: Without Nested Operation

ligent Transportation System communications (ITS), one of the most important works under this topic is the ISO TC204 WG16 or CALM[3] architecture. The scope of CALM is to provide a Standardized set of air interface protocols and parameters for medium and long range, high speed ITS communication using one or more of several media, with multipoint and networking protocols within each media, and upper layer protocols to enable transfer between media. Its key points are support of horizontal and vertical handovers with media selection and switching. This architecture supports several modes such as Car to Car and Car to Infrastructure communications.

- Proof of concept of the CALM architecture can be found on EU founded projects such as CVIS[4] or ANEMONE[5].
- Other NEMO usage scenarios are related with Personal Area Networks (PAN) communication. The most prominent project of this kind is Nautilus6[6]. This project is focused on researching new mobility solutions such as NEMO, MIPv6, Fast MIPv6..
- Finally some implementations and testbed have been done under the umbrella of projects such as, Daidalos[7] (creating a new architecture for the deployment of location independent, optimized personal services), e-Wheelchair (for people with disabilities and the elderly, providing health monitoring and remote communication with the wheelchair) or e-Bycicle (for monitoring of the performance/health condition of the cyclist).

More information related to research projects using NEMO can be found on [6].

3. REFERENCE DEVICE

As previously stated, the main aim of this work is to develop a NEMO implementation suitable to run on low end devices. In order to make this possible, several design decisions were made at early stages of the development:

- The implementation must be done on Linux. This allows the implementation to be compatible with a broad selection of devices instead of using some proprietary operating system which only can be use on a specific brand of devices.
- The devices running NEMO should be small, with low power profile, and will certainly have constrains on CPU power.
- The implementation must be based on common Linux functions, without any device specific optimization. This allows the implementation to be used on different devices.

Taking into account the design decisions and the price and size constrains imposed by the vehicular scenario we considered the use of a LinkSys WRT54G operating under the OpenWrt operating system as the reference device. On the following section we resume the characteristics of this device.

3.1 LinkSys WRT54G

On this section a brief summary of the LinkSys WRT54G is presented. The LinkSys WRT54G is one of the wireless routers with higher penetration into the market nowadays. It is based on a MIPs CPU running at 125 MHz, 16 MB of RAM and 4 MB of flash storage. In posterior versions it has been updated increasing the CPU speed up to 200 MHz and duplicating the memory, 32 MB of RAM and 16 MB of flash storage. The network part of this router is based on a 5 ports IEEE 802.3 switch, which are divided into 2 VLANs of 4 and 1 ports respectively (this last port is destined to be output to the Internet). Its Wireless capabilities

---

[3] www.calm.hu
[4] www.cvisproject.org
[5] www.ist-anemone.eu
[6] www.nautilus6.org
[7] www.ist-daidalos.org

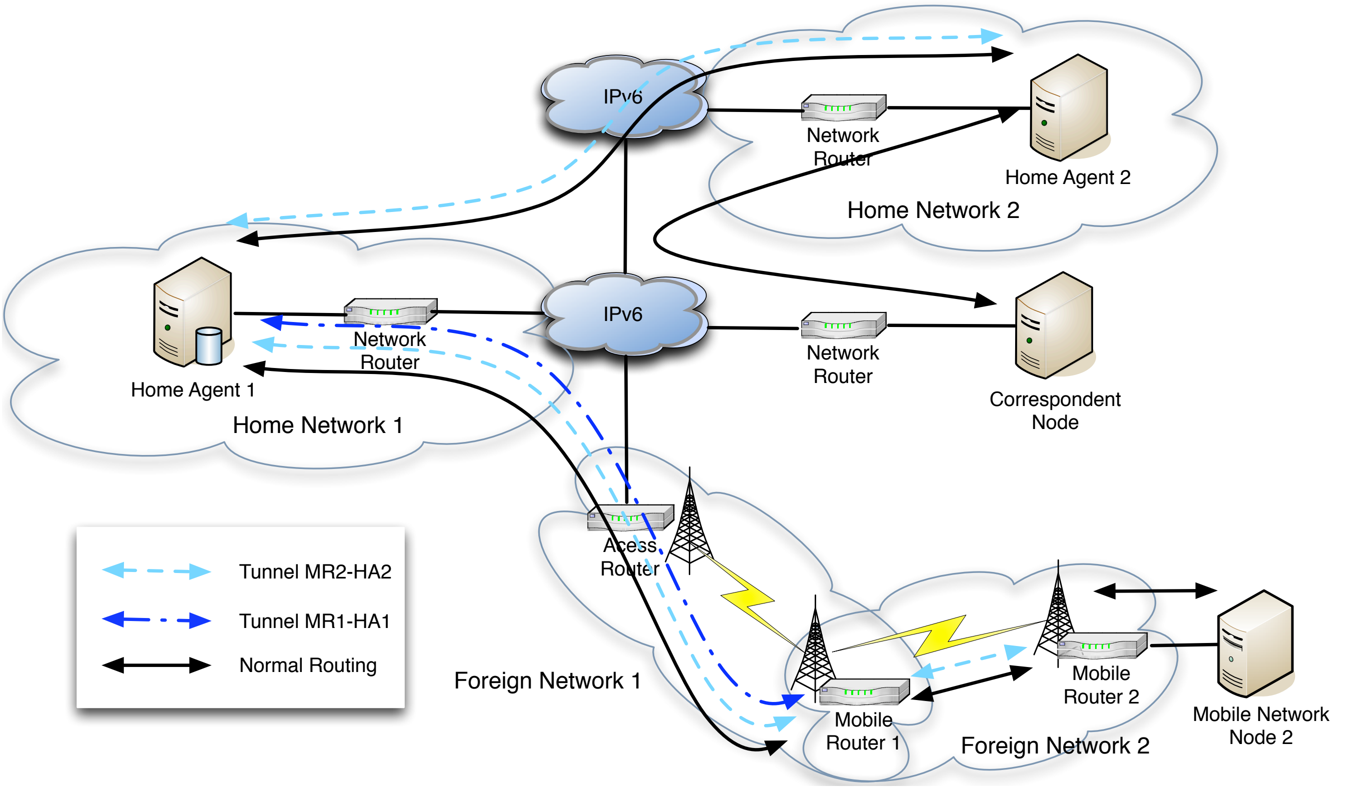

Figure 2: Nested Operation

are based on a Broadcom IEEE 802.11b/g chipset. Tables 1 and 2 present the technical characteristics of the Linksys WRT54G router. Analyzing these characteristics it can be seen that this router is very versatile and can be adapted to a huge range of conditions and uses. The set of networking characteristics is high enough to be integrated on vehicles, allowing wire and wireless connections. The dimensions are also small enough to be easily integrated on a vehicle, specifically in the porterage. This router can be modified on the software part, being able to run an Open Source operating system, designed by a community of developers and based on Linux. This software is called OpenWrt[8].

Table 1: Linksys WRT54G Specifications

Model Number:	WRT54G
Standards:	IEEE 802.3,IEEE 802.3u,IEEE 802.11bg
Channels:	11 Channels (US, Canada) 13 Channels (Europe) 14 Channels (Japan)
Ports/Buttons:	Internet One 10/100 RJ-45 Port LAN Four 10/100 RJ-45 Switched Ports One Power Port One Reset Button
Cabling Type:	UTP CAT 5
LEDs:	Power, DMz, WLAN, LAN (1,2,3,4)
,	Internet
RF Power:	Output 18 dBm
Security:	WAP, WEP, MAC filtering

Table 2: Linksys WRT54G Environmental Specifications

Dimensions:	7.32” x 1.89” x 7.87” W x H x D (186 mm x 48 mm x 200 mm)
Unit Weight:	17 oz. (0.48 kg)
Power External:	12V DC, 1.0A
Certifications:	FCC, IC-03, CE, Wi-Fi (802.11b,802.11g), WPA
Operating Temp:	32 °F to 104 °F (0 °C to 40 °C)
Storage Temp:	−4 °F to 158 °F (−20 °C to 70 °C)
Operating Humidity:	10% to 85% Non-Condensing
Storage Humidity:	5% to 90% Non-Condensing

3.2 OpenWrt Project

The WRT54G model is an unique example of a router designed for SoHo environments which runs on its commercial version Linux. This fact forced LinkSys to open its source code in order to comply with the GPL2.0 license included on the Linux SO. The openness of the source code has allowed developers along the world to modify the behavior of this router, adding new functionalities to it.

In January of 2004, the OpenWrt project began. Several developers on this project started implementing a new Open Source firmware for the WRT54G from scratch, during the years this firmware has been updated and ported to other platforms and manufacturers. OpenWrt is a GNU/Linux system optimized for routers, with an specific hardware and reduced capabilities. Its aim is to increase the functionality and performance of this kind of devices.

---

[8] www.openwrt.org

Currently, OpenWrt has two developing versions:

- WhiteRussian: This is the stable version of the system. It uses a Linux kernel 2.4.x and it only supports a reduced set of architectures.
- Kamikaze: This is the developing version, it supports Linux kernels 2.4 and 2.6. This version currently supports a broad range of architectures and devices. This is the version of the system used for the NEMO implementation.

4. IMPLEMENTATION DESIGN

We have developed an implementation of the NEMO Basic Support protocol [4] for low-end devices. It supports the movement of a MR to different foreign networks, working also with nested networks. The implementation has been developed for the Linux kernel 2.6.x branch. The NEMO Basic Support protocol is implemented in user space (signalling and movement detection and interface management) while routing and encapsulation is performed on the kernel. Through this design, the program is expected to work even when the kernel is upgraded, without requiring major changes. In fact the implementation is able to run on Linux boxes as well as over OpenWrt devices just recompiling it.
The implementation only supports implicit mode BUs. That is, the Mobile Router does not include a Mobile Network Prefix Option in the Binding Update. The Home Agent determines the Mobile Network Prefix(es) owned by the Mobile Router by manual configuration mapping to the Mobile Router's Home Address (HoA). A file is used to store these mappings. The software requirements are: a Linux device with kernel linux-2.6.x (tested with linux-2.6.22 and linux-2.6.19 on the LinkSys), support for IPv6-in-IPv6 tunnels enabled (used for the MRHA bidirectional tunnel).

4.1 MR operation

Movement detection is one of the main tasks of the Mobile Router. Mobile IPv6 does not impose any specific method to do that, but a simple movement detection mechanism is defined, based on IPv6 Neighbor Discovery [9]. This basically consists on listening to Router Advertisements (RAs). When the MR detects a new router advertising an IPv6 prefix different from its Home Prefix, the mobility management subroutine is launched.

At a first step, the routing table entries which correspond to the interface which has been moved, are deleted, because these routes are not useful anymore. All the routing table and interface's address modification is done using Netlink [5] sockets. By using this tool, we can manage the routing functionality of a Linux box by transferring information between kernel and user space. It consists of a standard sockets based interface for user processes, and an internal kernel API for kernel modules.

Afterwards the IPv6 address of the interface is removed and a new one is configured. This address is the CoA and is formed by the new prefix advertised (included in RAs) on the foreign link plus the EUI64 [7] of the interface. The EUI64 is built from the MAC address of the interface. Finally, a default route to the HA address, using the previously detected router on the new link as next hop, is inserted in the routing table.

After that, a BU must be sent to the HA informing of the current location of the MR (CoA). This BU is basically the same defined by Mobile IPv6, including a flag that indicates that it has been sent by a MR. Raw Sockets are used to send the signaling packets. By using this type of sockets we can build the entire IPv6 packet. We have followed this approach because normal sockets does not work well while changing the routing table and the interface address. The tunnel on the MR must not be created before a Binding Acknowledgment (BA) has been received, so the program waits for a BA arrival. When the BA arrives, it is processed and if everything is correct, the tunnel is set up. On the case of packet loss a retransmission of the Binding Update is performed. The ip6_tunnel module and a modified version of ipv6tunnel [1] are used for the creation, management and removal of IPv6-in-IPv6 tunnels.

In order to be able to reconfigure the MR's routing table when it comes back home, the routing table is stored.

While the MR is away from home and it is not moving among different visited foreign networks, it periodically sends BUs to refresh the binding between the MR's HoA and MR's CoA at the HA.

4.2 HA operation

The aim of this work is to perform an implementation of the Mobile Router functionality. The Mobile Router is the node which will travel on the vehicle and it has several constrains imposed by the specifics of the scenario. The Home Agent is an entity located on the Operator network and it does not have constrains imposed by the specific scenario we are studying. On our case the Home Agent has been inherited from a previous development performed on the Daidalos project. References to this implementation can be found on [3] and [2].

5. PERFORMANCE EVALUATION

The NEMO Basic Support protocol [4] provides transparent network mobility support, but presents some performance issues. The triangular routing phenomena due to the MR-HA tunnel adds processing delay and packet overhead. This problem is exacerbated when nesting is involved. The use of low-end device, which routing and processing power is very limited, increases the effect of this problems, reducing the available bandwidth of the end user. In order to experimentally evaluate the severeness of these problems, some practical tests and analytical studies have been performed. The experimental tests are focused on evaluating the final bandwidth obtained by the end user attached to the NEMO. We have chosen UDP for the bandwidth study since we wanted to measure the raw power of the low-end device in order to know if it is enough to be used on real applications.

5.1 Testbed Description

In order to test the correctness of our implementation, and perform some measurements to analyze the performance of the NEMO Basic Support protocol, a testbed was deployed. The structure of the testbed is shown in fig. 3. All the ma-

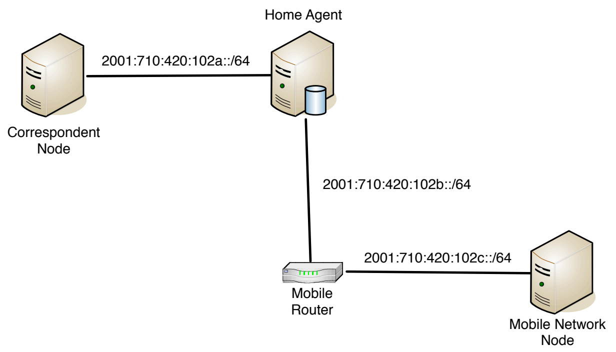

Figure 3: Testbed

chines are Linux boxes, with kernel linux-2.6.22, except the tests performed with the LinkSys which runs kernel linux-2.6.19. The routers are Linux boxes configured to this purpose. Our implementation is installed only in the HA and the MR. The traffic traces were collected at the Mobile Network Node and analyzed with specific purpose scripts. To generate the traffic, the mgen tool has been used. For each of the experiments, UDP traffic at a rate of 100Mbps were generated by the CN and sent to the Mobile Network Node. The NEMO protocol is very sensible to the delay between the CN, HA and MR. As we wanted to measure the routing raw power of the implementation, all elements which could add delay to the testbed were removed. Due to this reason, the CN is on the same network as the HA and the MR is also directly connected to the HA, although it is located on a foreign network. On this specific scenario, the routing capabilities of the low-end device can be measure, just being affected by the overhead in terms of bandwidth and processing time of the NEMO implementations.

5.2 Analysis of the Results

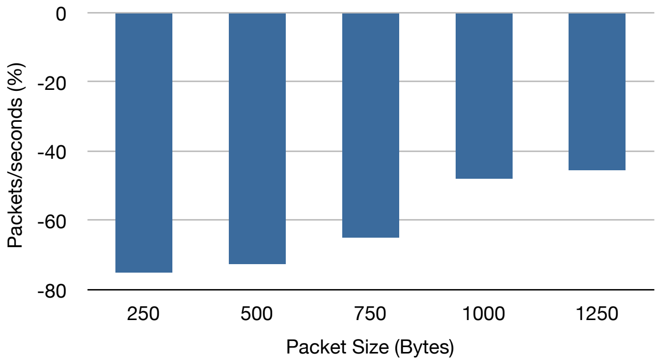

Figure 4: Relative performance comparison (LinkSys vs PC)

On this section a performance analysis by comparison of the developed implementation is performed. First the LinkSys WRT54G device is compared in terms of packets per second routed with a PC. This measurement allows us to put into a context the data extracted of the NEMO implementation performance on the device. On a second step we compare our NEMO implementation with a reference and well known

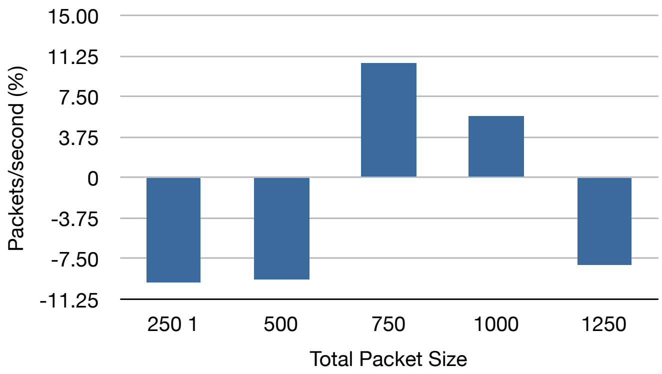

Figure 5: Relative performance comparison (NEPL vs NEMO) on PC

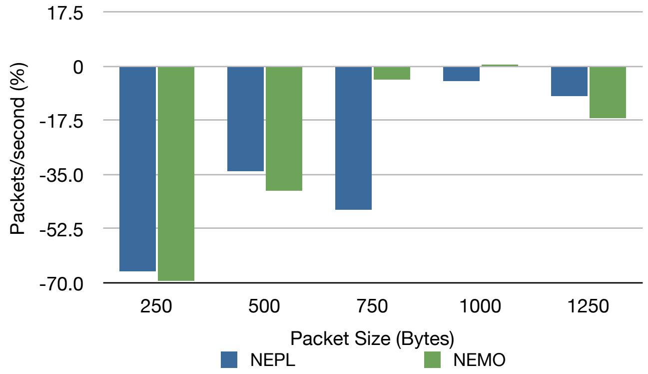

Figure 6: Relative performance comparison (Normal Routing vs NEMO vs NEPL) on PC

implementation, on this case the MIPL[9] implementation has been selected. By this comparison the design decisions taken on our implementation are validate. Finally we will present the performance achievable by our implementation running on the LinkSys device providing some bandwidth measures. In order to test the performance of the implementation, we have measured the amount of packets being routed on each of the following cases:

- Mobile Router Architecture PC, performing standard routing.
- Mobile Router Architecture PC, running the NEPL implementation.
- Mobile Router Architecture PC, running our NEMO implementation.
- Mobile Router Architecture LinkSys, performing standard routing.
- Mobile Router Architecture LinkSys, running our NEMO implementation.

Routers does not perform exactly on the same way with different packet sizes, for this reason we performed measures

---

[9] http://www.mobile-ipv6.org

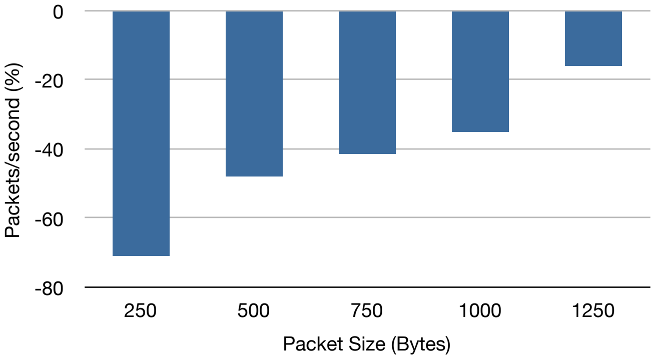

Figure 7: Relative performance comparison (Normal Routing vs NEMO) on Linksys

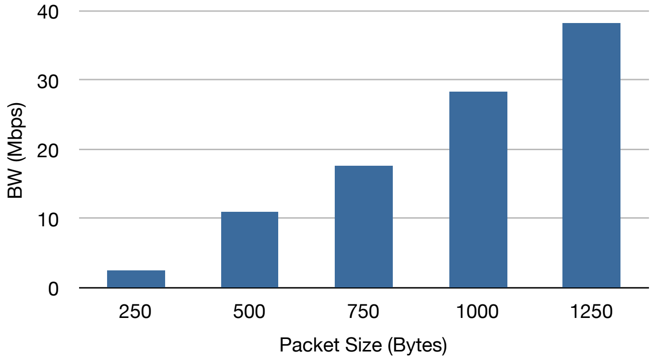

Figure 8: Bandwidth obtained using NEMO on the LinkSys architecture

for total packet sizes of 250, 500, 750, 1000 and 1250. The PC model used as MR and HA is a Intel P4 at 1500MHz with 1GB of RAM.

Figures 4, 5 and 6 present the comparison of performance done. Figure 4 shows the relative comparison of routing performance between the PC and a LinkSys WRTG54 router by presenting the relative decay in performance (%) of the routing process when the LinkSys router is used. As expected the processing power, hence the routing power, of the Linksys router is very reduced compared with the PC. The performance varies with the size of packets, decaying a 72% in the case of 500 bytes packets. The performance of the LinkSys router and the PC is not linear with the size of packets although it tends to be smaller when small packets are used, since as the packet size decrease, more packets arrive at the input queue of the router until it is overloaded.

Figure 5 presents the relative performance of our NEMO implementation compared with the NEPL. The 100% value is taken as the NEPL performance in packets/second for the current packet size, the bars indicate the difference in packets/second measured when our implementation is used. This figure shows how our implementation outperforms NEPL for some packet sizes and is outperformed by NEPL in others. The most important reduction in performance occurs for small packets. On this case the design of NEPL (performing some tasks on the Kernel) performs better than our implementation. The mean performance drop between both corresponds to our implementation being a 0.468% worst than MIPL. We argue that this small difference indicates that the implementation design in user space does not affect excessively the performance of the PC since the difference between the reference implementation and ours, in terms of performance is small.

Figure 6 presents the performance comparison between the NEPL implementation, the NEMO implementation for Lightweight devices and the standard routing on the PC architecture. The standard routing process on the PC is taken as 100%, and the bars represent the decrease in performance measured when using both implementations. It is clear that both implementations present an extra-load to the routing processing, being this very high when small packages are processed. As previously stated, the difference between our implementation and the reference implementation (NEPL) is not very high when medium size and big size packets are used. The performance of both implementations decreases when the packet size is smaller being our implementation worse in terms of performance than the NEPL.

The following step of this analysis corresponds to the measurement of the performance of the implementation running on the LinkSys router. Figure 7 presents the relative comparison between using standard routing in the LinkSys and using the NEMO implementation. The bars in figure 7 indicates the decrease in performance detected when using NEMO, being the 100% the standard routing value for this packet size. The results presented on this figure show how the LinkSys is highly impacted by the use of our implementation, being reduced its performance in a 40% in average. We argue that this loose of performance is due to the processing needed to encapsulate and de-capsulate the packets on the Mobile Router. As more packets are used to sustain the same binary rate, the number of encapsulation/decapsulation operations required grow. This is the reason of the decrease of a 70% of performance when using the smaller packet size. Note that the experiment takes into account the header overhead, the packet sizes have been changed per experiment, accordingly to the use of NEMO or normal routing.

Finally figure 8 presents the application bandwidth obtained by the Mobile Network Node while using the NEMO implementation on the Linksys. The bandwidth obtained reaches a peak of 38 Mbps, decaying with the size of the packet. The bandwidth obtained for the smallest packet size corresponds to 2.56 Mbps.

6. CONCLUSION

One of the main disadvantages of the NEMO protocol to be used as a basis for including vehicular communication technologies into commercial models is the size and price of the required equipment. Current devices used for this task are standard or reduced profile barebones, which although of small size, introduce a high cost into the vehicle. We think that the use of complete computers as Mobile Routers in the vehicles is not the best option. Instead we propose the use of low cost devices such as SoHo routers which are more suitable in cost and size terms. This devices use cheap processors which cannot be compared to the processing power of a standard computer. We argue this processing power is enough to run the NEMO protocol, and the performance of the devices is enough to provide communications inside a vehicle.

On this work we present a lightweight implementation of the Network Mobility (NEMO) protocol suitable to run on low end devices. In order to validate its performance several tests on a PC architecture has been done, comparing the results to the performance obtained by using standard routing and the NEPL implementation. Then, the implementation has been validated by using it on a LinkSys WRT54G domestic router.

The results obtained proved that the lightweight implementation performs reasonably well, with performance metrics similar to the NEPL implementation, which is used as reference implementation worldwide. When our implementation is used on a low-end device, on this case a LinkSys WRT54G, the performance obtained is low, but enough to be used as Mobile Router on cars or different vehicles. It is also important to note that the OpenWRT software used for the implementation is a beta version, we expect an increase in performance as the OpenWrt project is further developed.

7. REFERENCES

[1] Mobile IPv6 for Linux (MIPL).

[2] M. Calderon, C. Bernardos, M. Bagnulo, and I. Soto. *Securing Route Optimization in NEMO*, 2005.

[3] M. Calderon, C. J. Bernardos, M. Bagnulo, I. Soto, and A. de la Oliva. *Design and Experimental Evaluation of a Route Optimisation Solution for NEMO*, October 2006.

[4] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963, June 2004.

[5] G. Dhandapani and A. Sundaresan. *Netlink Sockets, Overview.* http://qos.ittc.ukans.edu/netlink/html/.

[6] T. Ernst. Ipv6 network mobility (nemo) and its usages.

[7] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. RFC 2373, July 1998.

[8] D. Johnson, C. Perkins, and J.Arkko. Mobility Support in IPv6. RFC 3775, June 2004.

[9] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, December 1998.