PRACTICAL EVALUATION OF A NETWORK MOBILITY SOLUTION

Antonio de la Oliva, Carlos Jesús Bernardos and María Calderón
Universidad Carlos III de Madrid
Avda. Universidad, 30 28911 Leganés
Madrid (SPAIN)
{ aoliva,cjbc,maria } @it.uc3m.es

Abstract   As the demand of ubiquitous Internet access and the current trend of all-IP com-
munications keep growing, the necessity of a protocol that provides mobility
management increases. The IETF has specified protocols to provide mobility
support to individual nodes and networks. The Network Mobility (NEMO) Ba-
sic Support protocol is designed for providing mobility at IP level to complete
networks, allowing a Mobile Network to change its point of attachment to the
Internet, while maintaining ongoing sessions of the nodes of the network. All
the mobility management is done by the mobile router whilst the nodes of the
network are not even aware of the mobility.

   The main aim of this article is evaluating the performance of the NEMO
Basic Support protocol by using our implementation.We also discuss the design
of an implementation of the NEMO Basic Support protocol.

1.   Introduction

   We are witnessing how the number of devices that are connected to the
Internet through wireless devices is continuously growing. The forthcoming
4G is expected to bring a new generation of portable, always-connected devices
that are able to connect to the Internet through heterogeneous technologies.

   Current standard Internet protocols, such as IPv4, do no support transparent
mobility. This is because 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 [Droms, 1997]
enable *portability* (i.e., a terminal can change its point of attachment and ob-

tain connectivity, but all its connections should be restarted). 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. Mobile IP [Johnson et al., 2004] is an IETF protocol designed to enable node mobility.

Nevertheless, supporting the movement of nodes is not the only problem that should be faced. Ubiquitous networking is becoming more and more common, so it is expected to have not only mobile nodes but also mobile networks. When the mobility problem is extended to support the movement of a complete network, we found that mobility management has to be provided to every device inside the network, even if these devices cannot afford the computational load that mobility implies. To provide mobility support to these networks, a protocol based on Mobile IP has been designed. This protocol is called Network Mobility (NEMO) Basic Support protocol [Devarapalli et al., 2004].

We have implemented in Linux a first prototype of the NEMO Basic Support protocol. This implementation has allowed us to study the performance of the protocol in several scenarios. Both the design of the implementation and a performance evaluation of the protocol (using the prototype) are presented in this paper.

The paper is structured as follows. A brief introduction to the basic concepts of Mobile IP and the NEMO Basic Support protocol are introduced in section 2. Section 3 explains the design of the implementation, detailing all the functionality, the tools used and the structure of the implementation. An analysis of the NEMO Basic Support protocol performance is presented in section 4, describing the testing scenarios used, in section 4.1. Finally, some conclusions and future work, are presented in section 5.

2.   Background

Providing mobility at IP-level is difficult because, as we have pointed previously, 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. A new Working Group (WG) within the IETF called NEMO was created to deal with the problem of complete networks that move as a whole. This WG has defined an extension of Mobile IP: the NEMO Basic Support protocol [Devarapalli et al., 2004], that enables Network Mobility support (see [Ernst and Lach, 2004] for the terminology).

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 where it resides when it is not moving. 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 [Devarapalli et al., 2004] 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 (fig. 1(a)).

This basic solution is derived from the solution proposed for host mobility support, MIPv6 [Johnson et al., 2004], without including the Route Optimisation support. Actually, the protocol is similar and the mobility signalling (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 (fig. 1(b)).
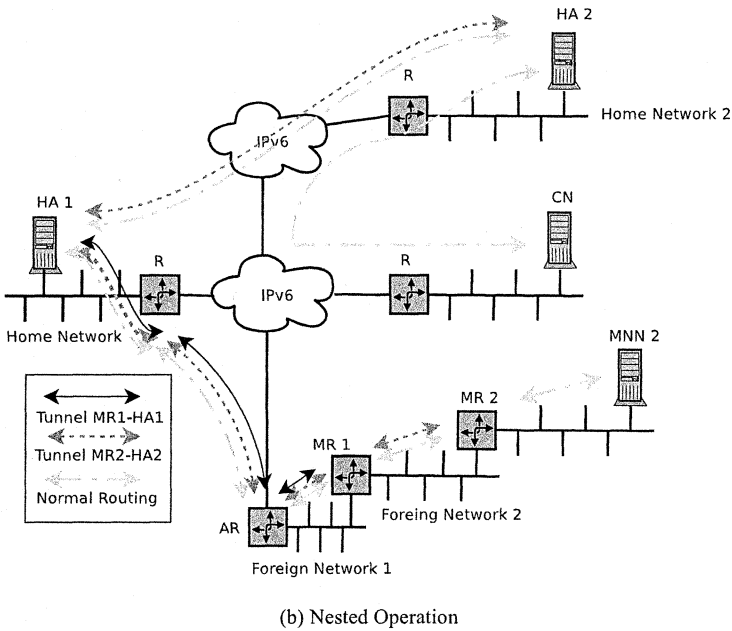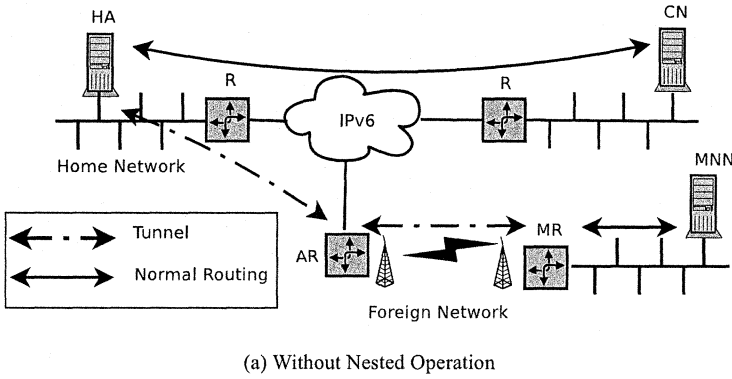
(a) Without Nested Operation

(b) Nested Operation

Figure 1.   Example of NEMO Basic Support protocol operation

3.   Implementation design

We have developed a first implementation of the NEMO Basic Support protocol [Devarapalli et al., 2004]. It supports the movement of a MR to different foreign networks, working also with nested networks. Besides the protocol operation, we have also implemented a tool for displaying the information stored in the different data structures that the MR and HA maintain (making easier

the debugging of the implementation). The implementation has been developed for the Linux kernel 2.6.x branch.

The NEMO Basic Support protocol is completely implemented in user space, because in this way the development is easier and quicker than doing that in the kernel. In addition, the program is expected to work even when the kernel is upgraded, without requiring major changes. Nevertheless, some kernel support is needed in order to recognise the messages introduced by the protocol. Otherwise, the kernel would send ICMPv6 [Conta and Deering, 1995] error messages triggered by the reception of one of those signalling messages. Basically, the kernel should identify the BU and Binding Acknowledgement (BA) messages and their reception should not trigger any special treatment, as they are processed by the user space implementation of the NEMO Basic Support protocol. This is done by modifying the kernel code in the IPv6 stack that process the IPv6 extension headers. This has been implemented in kernel linux-2.6.7, but should work without major changes in any kernel linux-2.6.x (a patch with these minor modifications has been created).

A single program implements both HA and MR functionalities, being selected by a flag in the configuration file.

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 machine with kernel linux-2.6.x (tested with linux-2.6.7), support for IPv6-in-IPv6 tunnels enabled (used for the HA-MR bidirectional tunnel) and the pcap (`http://www.tcpdump.org/`) library (used for the capture and processing of the mobility related signalling).

3.1   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 Neighbour Discovery [Narten et al., 1998]. This basically consists in 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 [Dhandapani and Sundaresan, 2005] sockets. By using this tool, we can manage the routing functionality of a Linux box by transferring informa-

tion 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 [Hinden and Deering, 1998] 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 signalling 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 must not be created before a BA has been received, so the program waits for a BA arrival. The program uses the capturing methods of the pcap library to wait for a BA reception. When the BA arrives, it is processed and if everything is correct, the tunnel is set up. The ip6_tunnel module and a modified version of ipv6tunnel (from `http://www.mipl.mediapoli. com/`) 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.

3.2   HA operation

The HA waits for the reception of BUs that indicates that a MR has moved. Again, the pcap library is used to capture these packets and retrieve the required information. When a BU is received, it is processed, checking if it fits in one of the following categories: a new BU from a MR that was at home (a new binding has to be created), a BU from a MR that was already away from home but has moved again (an existing binding has to be changed), a BU from a MR that is not at home and refreshes its binding information (the lifetime and sequence number of an existing binding has to be updated) or a de-registration BU (an existing binding has to be removed because the MR is again at home).

When a BU indicating that a MR has moved away from home is captured, several configuration steps are performed. First, the configuration file containing the HoA-MNP bindings is searched for the HoA included in the received

BU. If it is found, a new entry is added to the HA's Binding Cache (BC)[1]. Then the routes to the MR's MNP are removed from the HA's routing table. A new bidirectional IPv6-in-IPv6 tunnel is set up, and a default route to the MR through this tunnel (i.e., using the newly created tunnel interface) is added. After that, a BA is sent to the MR.

When the received BU is one refreshing an existing binding, the lifetime field and sequence number of the BC entry are updated accordingly. If no BU is received refreshing an existing binding before the lifetime expires, the BC entry is removed and the original configuration is reestablished.

Finally, if a de-registration BU is captured, the BC entry is removed and the original configuration is reestablished.

A visualisation tool, used to make easier the development, debugging and use of the implementation has been also developed. This tool, called *nemodiag*, prints the information of the MR's Binding Update List (BUL) or the HA's BC on the screen. To facilitate the development of this application and the management of these tables by the processes that refresh them, we have used shared memory.

4. Performance Evaluation

The NEMO Basic Support protocol [Devarapalli et al., 2004] provides transparent network mobility support, but presents some performance issues. The triangular routing phenomena due to the MR-HA tunnel adds delay and packet overhead. This problem is exacerbated when nesting is involved. 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 how the delay introduced by the triangular routing affects to the performance of applications. We have chosen TCP as the protocol to be studied, because it is representative of most of the traffic exchange in the Internet. Besides, the end-to-end delay affects the effective throughput of TCP applications. The analytical studies are focused in the packet overhead effects on several kinds of data traffic.

4.1 Testbed Description

In order to test the correctness of our implementation, and perform some measurements to analyse the performance of the NEMO Basic Support protocol, a testbed was deployed. The testbed is shown in fig. 2.

---

[1]The HA maintains a data structure where the information about the association between the MR's HoA and CoA is stored. The BC stores the HoA, the CoA, the sequence number of the last BU received and the lifetime of the binding.
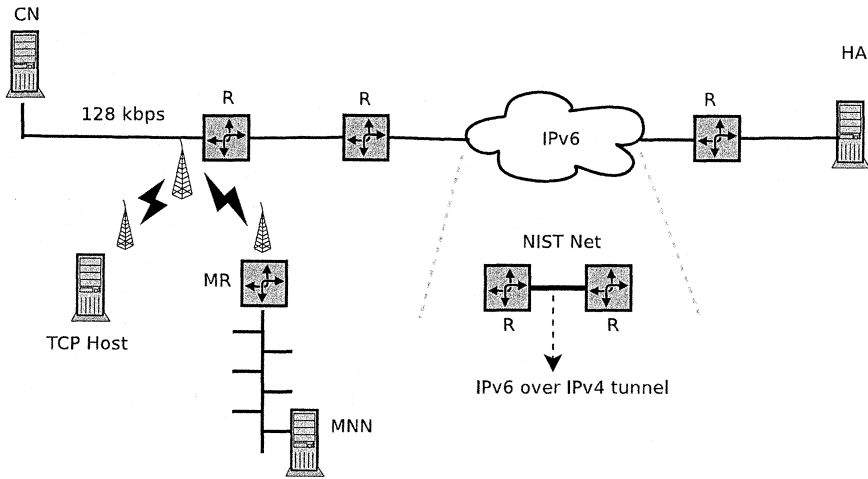
Figure 2.   Testbed

All the machines are Linux boxes, with kernel linux-2.6.7, except two inter-
mediate routers that run linux-2.4.22. The routers are Linux boxes configured
to this purpose. Our implementation is installed only in the HA and the MR.

In order to simulate a real IPv6 network between the HA and the network
that the MR is visiting, and be able to modify some characteristics of this
traversed network (e.g., latency, bandwidth, etc) the NIST Net network emula-
tor tool (http://www-x.antd.nist.gov/nistnet/) was used. This
tool was also used to restrict the bandwidth of the path between the CN and
the network that MR is visiting. NIST Net allows a single Linux PC, set up as
a router, to emulate a wide variety of network conditions. This software runs
only in IPv4 and with linux-2.4.x kernels. Therefore, in order to use it in our
testbed, we had to set up IPv6-in-IPv4 tunnels to use it. The use of tunnels
does not have any important effect on the tests, as just a very small delay due
to the encapsulation is added, and actually it reflects the real situation of the
IPv6 deployment, with several IPv4 clouds.

The traffic traces were collected at the CN and analysed with tcptrace (http:
//jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html).
Tcptrace is used to analyse the data collected and generate TCP graphics and
statistics about the traffic.

4.2   Effect of the Delay

In addition to the obvious effect that the delay has on performance (e.g., on
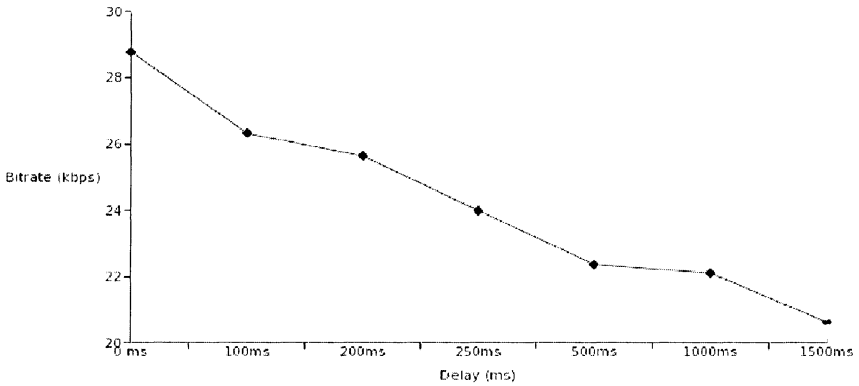real time applications), there is another aspect we have to take into account

Figure 3. MR's operation: Throughput vs Added Delay

when TCP is used. TCP throughput is affected by the end-to-end delay, therefore when a physical communication channel is shared between several TCP flows, the ones that have lower RTTs obtain a higher throughput.

In order to study the effect of the network mobility support on TCP, we have performed tests setting the bandwidth between the CN and the visited network to 128kbps (using the NIST Net emulator). Besides, the delay between the visited network and the Home Network is also varied by using the NIST Net emulator. This scenario allows us to make several tests with different delays, simulating different *distances* (i.e., RTTs) between the Home Network and the visited network.

The test basically consists in obtaining traffic traces on the CN, when a MNN is downloading a file located at the CN, while another computer attached to the same visited network downloads the same file four times simultaneously. All the five TCP flows share the available bandwidth (limited to 128kbps), and only the TCP flow sent to the MNN is affected by the introduced delay. This allows us to evaluate how the different delays and the NEMO Basic Support protocol affects the overall performance perceived by users of the TCP applications.

The effect of a higher delay in the TCP application is clear: the effective throughput decreases as the delay increases. Therefore, the available throughput for the other TCP flows (that have smaller RTTs) is bigger than the throughput of the MNN, because of the added delay introduced by the triangular routing.

A graph displaying the mean throughput of a MNN downloading a file located at the CN for different delays in the HA-MR path is shown in fig. 3. These results are as expected, showing how the delay affects the TCP throughput when several sessions are sharing the same physical bandwidth.

Traffic Type	Packet size (bytes)	Description
TCP	40	Minimum TCP packet size.
TCP	552	TCP without path MTU discovery
TCP	1500	Maximum Ethernet payload
UDP VoIP GSM	33	UDP-RTP packets coded with GSM
UDP VoIP G723.1	20	UDP-RTP packets coded with G723.1
UDP VoIP G711	240	UDP-RTP packets coded with G711
UDP VoIP LPC10	7	UDP-RTP packets coded with LPC10
UDP VoIP iLBC 20ms	38	UDP-RTP packets coded with iLBC
UDP VoIP iLBC 30ms	50	UDP-RTP packets coded with iLBC

Table 1. Traffic Types

4.3 Effect of the Packet Overhead

The use of the NEMO Basic Support protocol also increases the packet over-head, because of the tunnelling used between the HAs and MRs (this effect is even worse when nesting is involved). In this section we briefly study the over-head introduced by the NEMO Basic Support protocol, by comparing it with the overhead produced by plain IPv4 and IPv6.

The NEMO Basic Support protocol introduces a 40-byte extra IPv6 header to each packet in the HA-MR bidirectional path. Nesting introduces a 40-bytes extra headers for each level of chaining. The different types of traffic analysed are summarised in the table 1. Fig. 4 shows graphically the packet overhead for different kinds of IP payloads.

Nowadays, approximately the 83% of the total Internet traffic is TCP and other 13% is UDP [McCreary and K.Claffy, 2000], so fig. 4 is a good repre-sentative of how the NEMO protocol and the use of nested networks (up to 3 levels) would affect to most of the traffic that traverses today the Internet.

VoIP traffic is characterised for using small payload sizes, so the overhead in VoIP packets is very severe. This is specially important because of the expected use of VoIP in the forthcoming 4G communication networks.
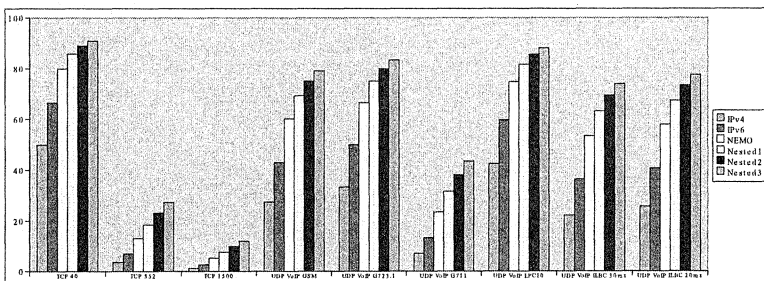

Figure 4. Overhead percentage by traffic types

Fig. 4 shows that even for payloads as small as 552 bytes, the overhead is not worthless (approximately 16%). When the number of nesting levels increases, the overhead grows dramatically. The overhead for 40-byte TCP packets (e.g., ACKs), is of a 80% without nesting; a 2-level nesting increases the overhead up to the 90%. These results show the necessity of a route optimisation mechanism for NEMO. The impact of using the NEMO Basic Support protocol for VoIP communications is very high (60% of overhead in average), so special care, and special design decisions, have to be taken if the NEMO Basic Support protocol (without using any route optimisation) is going to be used in VoIP networks.

To exemplify the impact of this packet overhead, we use the iLBC [Andersen et al., 2004] codec, used by skype (http://www.skype.com/) software. The iLBC (internet Low Bitrate Codec) is a free speech codec suitable for robust voice communication over IP. The codec is designed for narrow band speech and results in a payload bit rate of 13.33 kbps with an encoding frame length of 30 ms and 15.20 kbps with an encoding length of 20 ms. Eq. 1 and Eq. 2 are the analytical expressions of the packet overhead (iLBC payload over UDP/RTP) when IPv6 (Eq. 1) or the NEMO Basic Support protocol[2] (Eq. 2) are used. Table 2 summarises the bitrate needed to use the iLBC codec (over UDP/RTP) with IPv4, IPv6 and the NEMO Basic Support protocol (with different levels of nesting). It should be noted that a typical 64 kbps connection would be unable to support a VoIP communication of a 2-level nested mobile network. This supports our argument that route optimisation mechanisms for NEMO, are needed.

$$\frac{(40(IPv6) + 8(UDP) + 12(RTP) + 38(iLBC)) * 8\frac{bits}{byte}}{0.020} \tag{1}$$

$$\frac{(NL * 40(NEMO) + 40(IPv6) + 8(UDP) + 12(RTP) + 38(iLBC)) * 8\frac{bits}{byte}}{0.020} \tag{2}$$

	IPv4	IPv6	NEMO (no nesting)	NEMO (NL=2)	NEMO (NL=3)
BR (kbps)	31.2	39.2	55.2	71.2	87.2

Table 2. iLBC bitrates

5.   Conclusions and future work

The demand of ubiquitous Internet access (e.g., in public transportation systems) is increasing. Therefore, mechanisms that enable complete IP networks to be mobile without breaking ongoing connections of the nodes of the network are needed. The IETF NEMO WG has come up with an IP-level network

---

[2]NL=Nesting Levels. L=1 means a single mobile network, L=2 means 2 nested mobile networks and so on.

mobility solution: the NEMO Basic Support protocol, that enables a network to change its point of attachment.

In this paper we have developed an implementation of the NEMO Basic Support protocol for Linux, and we have used that to experimentally evaluate the performance of the protocol. The NEMO Basic Support protocol basically consists in setting a bidirectional tunnel between the MR and its HA. This tunnel adds both end-to-end delay and packet overhead. This delay can be unsuitable for some real-time applications, but also affects the overall TCP performance, as it has been practically showed in this paper. Besides, the added packet overhead increases the bandwidth requirements for applications. As an example, typical 64 kbps links would not be able to handle VoIP Skype calls of a node belonging to a two-level nested network.

The aforementioned inefficiencies need to be mitigated in order to facilitate the deployment of mobile networks. Therefore, work in route optimisation solutions would be a hot research topic in the near future.

Acknowledgments

References

Andersen, S., Telio, A. Duric, Astrom, H., Hagen, R., Kleijn, W., and Linden, J. (2004). Internet Low Bitrate Codec. RFC 3951.

Conta, A. and Deering, S. (1995). Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6). RFC 1885.

Devarapalli, Vijay, Wakikawa, Ryuji, Petrescu, Alexandru, and Thubert, Pascal (2004). Network Mobility (NEMO) Basic Support Protocol. RFC 3963.

Dhandapani, Gowri and Sundaresan, Anupama (2005). *Netlink Sockets, Overview.* http://qos.ittc.ukans.edu/netlink/html/.

Droms, R. (1997). *Dynamic Host Configuration Protocol.*

Ernst, T. and Lach, H-Y. (2004). Network Mobility Support Terminology. draft-ietf-NEMO-terminology-02.txt.

Hinden, R. and Deering, S. (1998). IP Version 6 Addressing Architecture. RFC 2373.

Johnson, D., Perkins, C., and J.Arkko (2004). Mobility Support in IPv6. RFC 3775.

McCreary, S. and K.Claffy (2000). *Trends in wide area IP traffic patterns - A view from Ames Internet Exchange.* CAIDA, Tech. Rep.

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