Performance Evaluation of NEMO Basic Support Implementations

Romain Kuntz, Koshiro Mitsuya, Ryuji Wakikawa

Keio University, Japan
E-mail: {kuntz,mitsuya,ryuji}@sfc.wide.ad.jp

Abstract. The NEMO Basic Support protocol was standardized to provide network mobility and has been implemented by several organizations. Though interoperability among those implementations has been confirmed, performance evaluation is one of the important next step to expect deployment in real environments. This paper reports the results of interoperability and performance tests conducted on NEMO Basic Support implementations. The performance tests are targeting the Mobile Routers and are represented with UDP and TCP throughput, Round Trip Time and Handover latency. Although more tests need to be performed to conform to real-life usages, the basic features of the implementations have been evaluated.

1 Introduction

In order to fulfill the demand for on-the-move and uninterrupted Internet connectivity in Mobile Networks (groups of nodes moving together), the Internet Engineering Task Force (IETF) has standardized the NEMO Basic Support protocol [1] in January 2005. As of today, several organizations are developing their own NEMO Basic Support implementation on different platforms [2].

Validation of an implementation is an important step before the deployment of the technology. We cannot expect from people to use an implementation as long as it has not been proved that it is complying with the specifications and competitive enough to support high loads.

The TAHI project [3] provides conformance test tools for NEMO Basic Support implementations. However, no automatic tools to evaluate the performance of a Mobile Router exist yet. We thus have performed the evaluation of some Mobile Routers to investigate possible bottlenecks and problems in the implementations or the protocol specification.

We first give an overview of each implementation involved in our tests. We report the results of the interoperability tests conducted between those implementations. We then explain our evaluation scenario and report the results we obtained. We finally compare and analyse further those results.

2 Implementations Overview

In this paper, two NEMO implementations, namely NEPL and SHISA, were used to evaluate interoperability and performance.

NEPL 0.1.1 (NEMO Platform for Linux) is implemented as an extension of MIPL2 (Mobile IPv6 for Linux [4]) on the Linux 2.6.8.1 kernel. It has been developed by Go-Core project at the Helsinki University of Technology in cooperation with Nautilus6 project [5] from WIDE [6]. SHISA [7] is an implementation of Mobile IPv6 and NEMO Basic Support on BSD operating systems. It has been developed by KAME [8] and Nautilus6 projects in WIDE. NEPL and SHISA support both implicit and explicit binding registration modes for NEMO Basic Support and DHAAD to discover NEMO-capable Home Agents. SHISA also supports various extensions such as Multiple Care-of Addresses Registration [9]. Those implementations are freely available but still work in progress.

3 Interoperability Tests

Our test environment is heterogeneous, with a Home Agent and a Mobile Router using different implementations. We thus have conducted several tests to confirm the interoperability:

- Mobile Router registration and deregistration to the Home Agent, using explicit mode and implicit mode, and using DHAAD to discover the Home Agent address.
- Communications between two Mobile Networks, with the same or a different Home Agent for each Mobile Router.
- Up to two levels of Nested mobility, including Nested Mobile Networks and Visiting Mobile Node in a Nested Mobile Network.

Those tests have been conducted at the 6th TAHI interoperability event in January 2005, involving NEPL, SHISA and the Cisco implementation (experimental version). We have successfully confirmed the interoperability following the scenario detailed in the report [10].

4 Performance Evaluation

We show the parameters of the experimental network and the results obtained from the evaluation. We then present an analysis of those results in section 4.8.

4.1 Testbed Topology

The experimental testbed (Fig. 1) is composed of three routers and one client node.

HA is a Linux router with a 2.6.8.1 kernel. It can also act as a Home Agent. In order to keep the same test environment while operating NEPL or SHISA on the Mobile Router, the Home Agent operates NEPL 0.1.1 for all the tests.

The router R is running Linux and connecting the Foreign Link 1 (FL1) to the Foreign Link 2 (FL2).

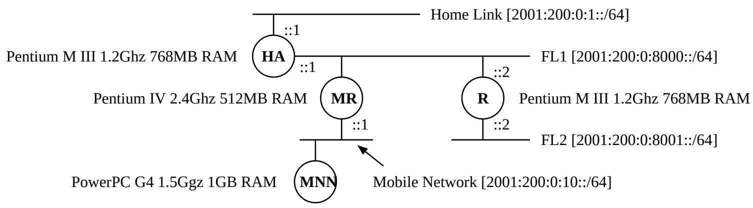

Fig. 1. The experimental test network

MR can run either FreeBSD 5.4 or Linux with a 2.6.8.1 kernel. It can also be configured as a Mobile Router running SHISA (snap 20051003 on FreeBSD 5.4), or NEPL 0.1.1 (on Linux with a 2.6.8.1 kernel). Mobile Nodes and Mobile Routers are often equipped with wireless media such as 802.11b and PHS. However, the measured data is not expected to be accurate due to wireless interferences. As we need high speed and reliable media with side effects reduced to the minimum for the performance tests, we use wired interfaces (Ethernet) to interconnect the equipments to the network. MR can thus connect on the Home Link, FL1 or FL2 using 100BaseTX full-duplex Ethernet. For the experimentation of NEMO Basic Support, MR is a Mobile Router and HA is a Home Agent. MR can connect to either FL1 or FL2 and register to the Home Agent. After a successful binding registration, all packets sent from the Mobile Network are encapsulated in an IPv6-over-IPv6 tunnel established between MR and HA.

MNN is a MacOS X 10.3.9 node. It is an usual IPv6 node located behind MR. It is also referred as a Mobile Network Node when MR is acting as a Mobile Router.

4.2 Test Parameters

The performance comparison quantitatively evaluates the overhead added by the NEMO Basic support implementations. In this paper, performances are based on the following parameters:

- UDP and TCP throughput: in our tests, these are the highest throughput that a router can reach by forwarding UDP datagrams or TCP packets. Higher throughput means better forwarding performance.
- Round Trip Time (RTT): this is a measure of the delay on a network between two nodes. In our tests, lower RTT means better processing time on the nodes in the path.
- Handover Latency: in our tests, this is the difference between the handover time on the Mobile Router and on the Mobile Network Node. The smaller the difference is, the better the routing reestablishment on the Mobile Router is.

4.3 UDP Throughput

Purpose of the test: The aim of this test is to compare the UDP throughput of a router to the one of a Mobile Router, in order to check the overhead the Mobile Router capability introduces.

MR is connected to FL1. It is thus directly connected to HA to avoid a bottleneck between those two nodes. HA is running an **iperf** [11] server in IPv6 and single threaded UDP mode. MNN is running an **iperf** client in IPv6 and UDP mode, and sends UDP datagrams with 1192 bytes as playload size (which ensures to be under the Path MTU, as explained in Section 4.8).

The test consists in finding the maximum throughput that can be sent from MNN to HA when MR is first acting as a router, then when MR is acting as a Mobile Router. We thus increase step-by-step the bandwidth of the UDP flow (with the **-b** option of iperf) sent from MNN to HA until we reach the maximum limit. The limit is known when the rate used to send the UDP datagrams from the iperf client is higher than the rate at which those datagrams are received on the iperf server.

Results: Results for both FreeBSD and Linux are shown on Fig. 2. We can notice that the maximum UDP throughput is lower when NEMO Basic Support is operated, for both NEPL and SHISA. When SHISA is operated, we can observe oscillations on the UPD throughput. Lower average UPD throughput (for the 91, 93, 95, 97 and 99 Mbps tests) are obtained when a binding update is sent during the test. Sending a Binding Update during a test results in a lower UDP throughput. Such behaviour does not appear when NEPL is operated.

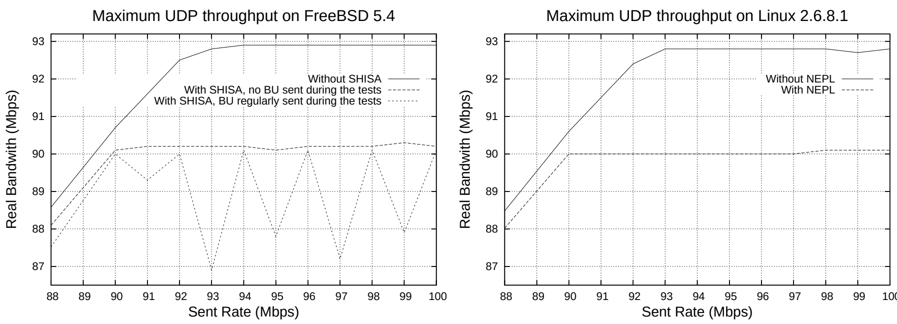

Fig. 2. UDP Throughput with and without NEMO Basic Support

4.4 UDP Throughput with Fragmentation

Purpose of the test: We execute an UDP throughput test using the maximum rate allowed by the router (92Mbps) and the Mobile Router (90Mbps), with the UDP packet size increasing and exceeding the Path MTU.

MR is directly connected to HA on FL1. HA is running an **iperf** server in IPv6 and single threaded UDP mode. MNN is running an **iperf** client in IPv6 and UDP mode. We increase step-by-step the length of the datagram's playload (with the **-l** option of iperf). This test is first executed with MR acting as a router, then as a Mobile Router.

Results: Results for both FreeBSD and Linux are shown on Fig. 3. The graph shows that the maximum avalaible bandwidth decreases from a certain payload size (displayed as the Path MTU cross). This Path MTU differs according to the implementations. When operating SHISA, we can notice that once the PATH MTU is reached, the throughput drops sharply and then constantly increases. It would be expected that the througphut decreases as the packet size increases.

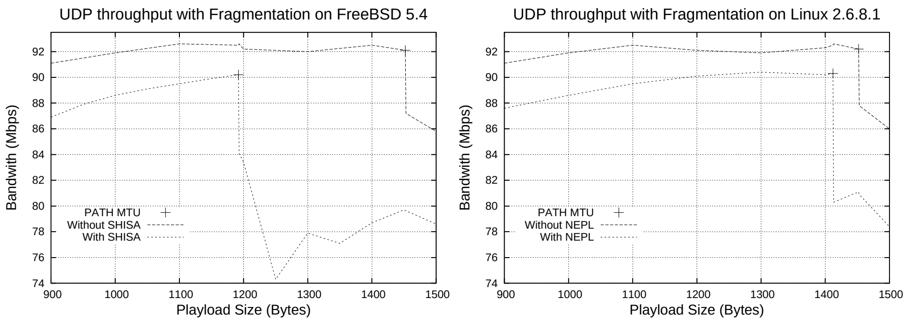

Fig. 3. UDP Throughput with Fragmentation, with and without NEMO Basic Support

4.5 TCP Throughput

Purpose of the test: The aim of this test is to compare the TCP throughput of a router to the one of a Mobile Router, in order to check the overhead the Mobile Router capability introduces.

MR is directly connected to HA on FL1. HA is running an **iperf** server and MNN an **iperf** client, in IPv6 and TCP mode. This test is performed when MR acts as a router and then as a Mobile Router.

Results: Results for both FreeBSD and Linux are shown on Fig. 4. We can see that the TCP throughput is lower when NEMO Basic Support is operated, for both NEPL and SHISA.

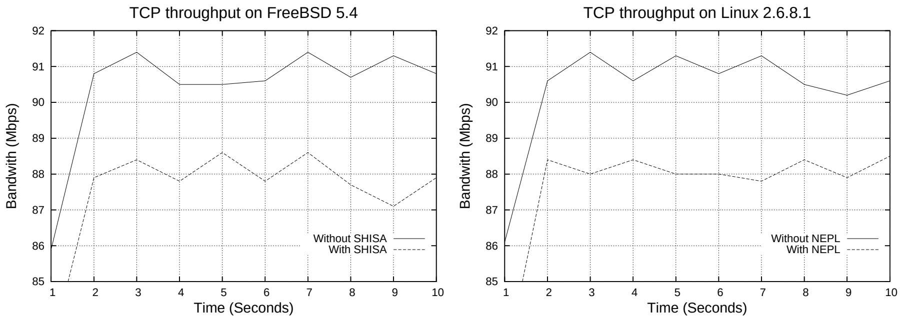

Fig. 4. TCP Throughput with and without NEMO Basic Support

4.6 Round Trip Time

Purpose of the test: The aim of this test is to compare the Round Trip Time (RTT) between MNN and HA when MR is acting as a router and then as a Mobile Router, in order to check the overhead the Mobile Router capability introduces.

MR is the only router located between MNN and HA. To calculate the RTT, we use the **ping6** command on MNN with HA as the target. We first calculate the average RTT with the default ICMP echo/reply packet size (64 bytes), then we increase step-by-step the playload size (with the **-s** option of ping6) until we exceed the Path MTU.

Results: The Table 1 shows that using NEMO Basic Support adds some delay in the round trip time between the MNN and its correspondent.

FreeBSD5.4	SHISA	Linux 2.6.8.1	NEPL
0.398	0.547	0.607	0.631

Table 1. Average RTT (ms) with 50000 ICMPv6 echo/reply

The Fig. 5 shows that the RTT increases in proportion to the playload size. We can identify thresholds (diplayed as the Path MTU cross) where the growth is bigger when Linux (with or without NEPL) is operated on the Mobile Router.

4.7 Handover Latency

Purpose of the test: The goal of this test is to compare the handover time on the Mobile Router MR and on the Mobile Network Node MNN, in order to check if being in the Mobile Network introduces a latency in the handover time.

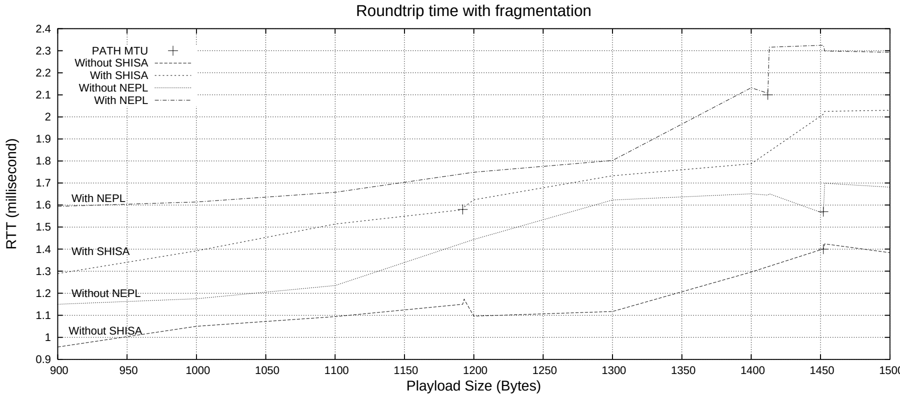

Fig. 5. RTT with Fragmentation, with and without NEMO Basic Support

A layer-1 switch allows the Mobile Router to switch almost instantaneously between the Home Link, FL1 and FL2. The delay introduced by this switch is maximum 1.28ms, and the jitter is about 100μs. This switch thus does not introduce any significant handoff delay.

We conduct the tests with ICMP traffic, using the **ping6** command at a high frequency: icmp echo packets are sent every tenth of a second. We perform three handovers during this test: from the Home Link to FL1, then from FL1 to FL2, then from FL2 to the Home Link.

Results: The Fig. 6 shows the handover latency between the handover time on MR and the handover time on MNN when SHISA or NEPL is operated on the Mobile Router. We do not note a significant difference between the handover time measured on each node.

4.8 Analysis

UDP and TCP throughput The difference we note in the TCP and UDP throughput when NEMO Basic Support is operated on the router can be explained by the tunnel overhead added by NEMO Basic Support between the Mobile Router and the Home Agent.

The extra IPv6 header for tunnel encapsulation adds a 40 bytes overhead in each packet. For example, the UDP throughput tests consists in sending 1192 bytes from MNN to HA. Adding the UDP and IPv6 headers results in a 1240 bytes datagram. The extra IPv6 header adds 40 bytes more which results in a 1280 bytes datagram. This overhead represents 3.1% of the total size of the datagram.

The maximum UDP throughput is about 92.8Mbps with both the Linux and the FreeBSD router. The maximum UDP throughput is 90.1Mbps with both

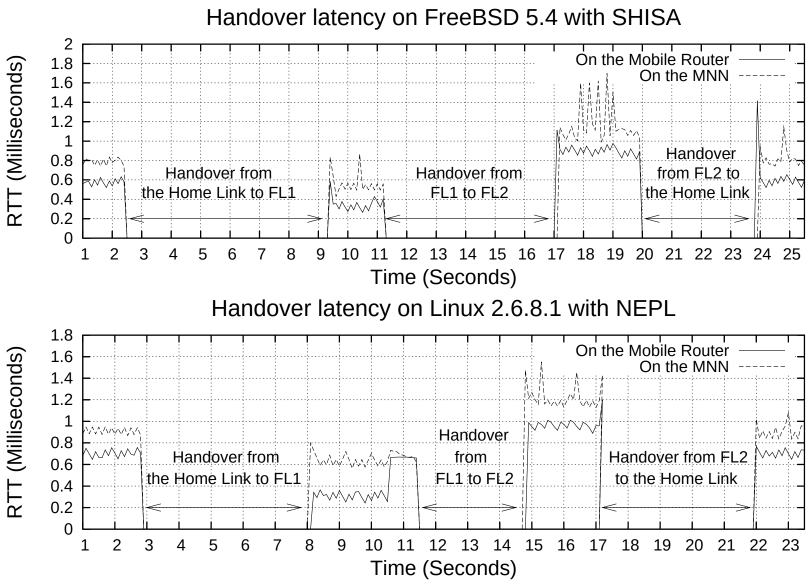

Fig. 6. Handover Latency on the MR and on the MNN

SHISA or NEPL operating on the Mobile Router, which represents a 2.9% difference with the result we obtain on the router. Taking into account the processing overhead, this confirms that the results we obtained are reasonable.

However, when SHISA is operated, sending a Binding Update results in a drop of the average UDP throughput. The binding registration procedure induces a state where packets cannot be sent in the MR-HA tunnel during a short time. This problem is known from the developers and currently investigated.

Roundtrip time The RTT increases when operating NEMO Basic Support on the Mobile Router. NEMO Basic support implementations use an IPv6-over-IPv6 tunnel between the Mobile Router and its Home Agent. Each IPv6 packet forwarded by the Mobile Router to the Internet, or by the Home Agent to the Mobile Network, is encapsulated in a new IPv6 packet. Each encapsulated packet received by the Home Agent and the Mobile Router from the tunnel is decapsulated before being processed. Such operations have a cost and can explain some of the roundtrip time difference we can observe on Tab. 1 and Fig. 5. However, more tests and a deeper analysis are necessary to conclude if such difference is acceptable or not.

UDP and RTT fragmentation Fig. 3 and 5 show that when the playload size of the packet sent through the Mobile Router or the router increases, it reaches at some point of time a value where the maximum UDP throughput suddenly decreases and the RTT increases. This threshold is called Path MTU, which is the minimum MTU on the path between MNN and HA.

The default MTU on FreeBSD 5.4 and on Linux 2.6.8.1 is 1500 bytes. By removing the size of the IPv6 header (40 bytes) and the size of the UDP header (8 bytes), we get 1452 bytes available for the playload. When operating SHISA, the default MTU is 1280 bytes for the IPv6-over-IPv6 tunnel. By removing the size of two IPv6 headers (because of the IPv6-in-IPv6 encapsulation) and the size of the UDP header, we get 1192 bytes available for the playload. When operating NEPL, the default MTU is 1500 bytes for the IPv6-over-IPv6 tunnel. By removing the size of two IPv6 headers and the size of the UDP header, we get 1412 bytes available for the playload. Those values match the Path MTU crosses on Fig. 3 and 5.

When the Path MTU is exceeded, all packets must be fragmented by the host that send them, and each packet must be rebuilt from the fragments by the correspondent host. Those operations have a cost and results in increasing the RTT. Also, for each datagram, two or more fragments are sent instead of one single packet. This results in a header overhead that explains the decrease in the maximum UDP throughput.

When operating SHISA, once the PATH MTU is reached, the throughput drops sharply and then constantly increases. Although we are not sure about the exact cause of this behaviour, we believe this is linked to PATH MTU discovery, which is not correctly handled when SHISA is operated on the Mobile Router. This problem is known from the developers and is being investigated.

5 Conclusion

Interoperability and performance tests of an implementation are an important part in the development process. This paper presents the tests performed against the NEPL and SHISA implementations, and analyses the results. In order to improve and enrich those results, more tests need to be done in the future, such as in nested or multihomed configurations.

Acknowledgement

We would like to thank Yoshihiro Onodera, Akiko Hattori and Pascal Thubert from Cisco Systems for their great help including discussions and equipments rentals for the interoperability testings.

References

1. Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert. NEMO Basic Support Protocol. Request For Comments 3963, IETF, January 2005.
2. Supplemental information web page of the IETF NEMO working group, as of December 2005. http://www.mobilenetworks.org/nemo/.

3. WIDE TAHI Project website, as of December 2005. http://www.tahi.org.
4. MIPL, Mobile IPv6 for Linux, as of December 2005. http://www.mobile-ipv6.org.
5. WIDE Nautilus6 working group website, as of December 2005. http://www.nautilus6.org.
6. WIDE Project, Widely Integrated Distributed Environment, as of December 2005. http://www.wide.ad.jp.
7. SHISA, an implementation of Mobile IPv6, as of December 2005. http://www.kame.net.
8. KAME Project, Free IPv6 and IPsec stack for BSD variants, as of December 2005. http://www.kame.net.
9. Ryuji Wakikawa, Keisuke Uehara, Thierry Ernst, and Kenichi Nagami. Multiple Care-of Addresses Registration. Internet Draft draft-wakikawa-mobileip-multiplecoa-04.txt, IETF, June 2005. Work in progress.
10. Romain Kuntz. NEMO Basic Support implementation tests at the 6th IPv6 TAHI Interoperability Test Event. Technical report, Nautilus6, February 2005. http://www.nautilus6.org/doc/tc-nemo-tahi-interop-20050207-KuntzR.txt.
11. Iperf, a tool to measure TCP and UDP bandwidth performance, as of December 2005. http://dast.nlanr.net/Projects/Iperf/.