# Multiple Mobile Routers in NEMO: How Neighbor Discovery Can Assist Default Router Selection

Romain Kuntz, Julien Montavont and Thomas Noël
LSIIT (UMR CNRS 7005)
Louis Pasteur University Strasbourg, France
kuntz@lsiit.u-strasbg.fr, {montavont,Thomas.Noel}@dpt-info.u-strasbg.fr

***Abstract*—The Network Mobility Basic Support (NEMO BS) protocol is the IETF standard to manage the mobility of entire IPv6 networks. One of the typical applications of this protocol is to deploy NEMO BS in transportations such as train or bus. As a result, passengers can benefit from global and permanent IPv6 connectivity with legacy IPv6 terminals equipped with a common wireless technology such as Wi-Fi. However, a mobile network is generally managed by a single mobile router which carries out all operations related to mobility and packet forwarding. In this article, we present a new proposal which enables the cooperation of multiple mobile routers to improve the bandwidth, network coverage and reliability of a mobile network. In addition, a dynamic load sharing mechanism between all available mobile routers is supported. This solution mainly relies on Neighbor Discovery and has been experimented on a real testbed.**

***Index Terms*—IPv6 Mobility, Network Mobility (NEMO) Basic Support, Mobility Management, Multihoming**

## I. Introduction

Rapid deployments of wireless technologies have stimulated the appearance of new types of user behavior and expectations. The concept of pervasive connectivity (i.e. being connected anywhere, anytime) is, among others, of crucial importance. The users expect to benefit from their usual network applications or services while on the move. However, layer 3 movements (i.e. when a roaming node moves across various IP networks) require the nodes to change their IP addresses to avoid routing issues. Without specific support, such IP address changes would break ongoing communication. In addition to the Mobile IPv6 protocol which is designed to enable host mobility across IPv6 networks, the IETF has recently defined the Network Mobility Basic Support (NEMO BS) protocol [1] which enables mobility of entire IPv6 networks. NEMO BS is especially expected to be deployed in transportations (public or personal) to offer global and permanent IPv6 connectivity to the passengers [2]. In NEMO BS, the mobility management operations are carried out by the mobile network router known as Mobile Router (MR). Legacy IPv6 nodes (i.e. without additional software) located in the mobile network are provided with a global IPv6 prefix which remains valid whatever the location of the mobile network. While on the move, the MR maintains an IPv6-in-IPv6 tunnel with a remote fixed node called the home agent. All network movements are therefore transparent to the nodes located in the mobile network, the MR and the home agent performing the necessary operations to route packets destined to or originated from the mobile network. Without considering the home agent, the reliability of the mobile network is directly dependent of the MR status. Whenever the MR fails (e.g. breakdown, lack of network coverage, etc.) the mobile network is disconnected from the Internet breaking all ongoing communication of the Mobile Network Nodes (MNN). Although the lack of coverage of a particular wireless technology could be overcame with a multi-interfaced MR using the Multiple Care-of Addresses (MCoA) extension [3], the mobile network is still subject to a failure of the MR itself. In case of an intense traffic, data packets could be potentially delayed or even dropped when forwarded by the MR to the Internet. In fact, the MR may be a bottleneck for the whole mobile network, as the available bandwidth inside is usually greater than the one offered by the access networks the MR connects to.

To address these issues, we could provide the mobile network with multiple MRs. In addition to increasing the overall mobile network bandwidth and coverage, the cooperation of multiple MRs could enable a failover mechanism between MRs. Furthermore, the overall load of the mobile network could be shared between all available MRs, resulting in a more robust system. In this article, we present the first components of a new protocol which enables the simultaneous use of multiple MRs in the same mobile network. Based on Neighbor Discovery [4], our solution allows dynamic discovery between MRs, load sharing between all available MRs and failover support. Furthermore, this solution does not require additional software on MNNs so that the entire system remains compliant with legacy IPv6 clients. Our solution has been evaluated through experiments performed on a real testbed using Scapy6 [5].

The rest of the document is organized as follows. After a brief overview of the existing solutions supporting multiple MRs, we detail in Section III our proposal and the targeted environment. Section IV presents the testbed that we have used for all experiments and provides an analysis of the obtained results. Finally, conclusions and future work are presented in Section V.

## II. Related Work

In order to support the management of multiple MRs in a single mobile network, four points have to be considered. First, the MNNs have to select a default exit router among multiple ones. Then, all the MRs need to synchronize some information

978-1-4244-2644-7/08/$25.00 ©2008 IEEE

such as their availability and characteristics. Each MR also has to inform its home agent about how the incoming traffic should be routed towards the potential multiple paths. Finally, the multiple home agents have to exchange information about the MR they manage. Research on NEMO BS is relatively recent and only few articles in the literature address the management of multiple MRs located in the same mobile subnet. Furthermore, most of them do not focus on all requirements suggested in [6].

A first proposal [7] presents a dynamic load sharing mechanism taking place on home agents. This proposal considers various MRs and home agents being held by several operators. Each MR in the mobile network can discover, authenticate and register a neighbor MR to its home agent. Each home agent can therefore maintain a list of alternative tunnels towards different home agents and MRs to reach its own MR. The home agent can then perform load sharing between the legacy NEMO BS tunnel and alternative tunnels using tunnel latency as a metric. However the authors do not detail how the traffic is really redirected (packet by packet, flow by flow, etc.). In addition, it is fairly improbable that various operators accept to relay (through their home agents) traffic sent to clients belonging to a competitor. Another proposal [8] presents a protocol focusing on reliability. In essence, the proposed solution allows a MR to act as a substitute for a failed one. Note that MRs may belong to different operators. Three error cases are supported: failure of the egress or ingress interfaces, and complete MR failure. Upon failure detection, a neighbor MR registers to the home agent of the failed MR in order to provide Internet connectivity to the MNNs on behalf of the failed MR. Although this solution presents an interesting MR redundancy mechanism, it requires to register the MR from one operator to the home agent of another operator. Finally, the proposal presented in [9] allows a mobile network to be served through multiple MRs transparently to the MNNs. Basically, all the MNNs are connected to a unique MR, known as the primary MR, and the other MRs are seen as virtual interfaces of the primary MR. All traffic from MNNs is therefore sent to the primary MR which forwards it among all the available interfaces (real or virtual) according to the installed routing policies and preferences. However, the transmission of a packet through a virtual interface would generate an overhead in the mobile network as such packet is encapsulated by the primary MR before being sent again on the mobile subnet to a non-primary MR. Such overhead may seriously degrade the quality of effective communications in case of a wireless mobile subnet.

On the other hand, there are several schemes addressing load sharing or redundancy between routers in fixed IPv6 networks such as [10]. However, these proposals are hardly suitable to mobile networks as they were not designed considering the problem in the mobility context.

## III. Overview of the solution

Our proposal focuses on multiple MRs located in the same mobile subnet. These multiple MRs are associated to one or multiple home agents and advertise the same mobile network prefix in the mobile subnet. This scenario refers to the (n,*,1) model as described in [6]. Though the mobile network of a vehicle might be divided in multiple subnets, it is likely that the user segment (where the MNNs are located) will be served by MRs that belong to the same operator. We believe that this model is one of the most practical ones [11], all MRs and MNNs being on the same layer 2 link, i.e. each node (either MR or MNN) could directly join any other node in the subnet.

In this paper, we focus on the assignment of a default router to the MNNs, i.e. through which router all the traffic from a MNN to the Internet is sent. We do not yet consider how the downstream flows can follow the path back to that very router. This problem is part of the interaction of the MR with its home agent and will be addressed in a future work. Note that we use advanced features of Neighbor Discovery, so readers may refer to [4] for further details on its related mechanisms.

### A. Discovery of Neighbor Mobile Routers

Before cooperating, all MRs of the mobile network have to discover their potential neighbor MRs. Each MR is preconfigured with a role: Master or Slave. The Master MR (which is unique in the mobile network) is in charge of the selection of the default MR for each MNN. In addition, the Master MR carries out the transmission of router advertisements over the mobile subnet. A Slave MR monitors the status of the Master and may react upon detection of Master MR failure (see Section III-C). A Slave MR could also be selected (by the Master) as the default router for a set of MNNs. Note that the roles could be assigned dynamically with a well-known and efficient election algorithm (e.g. the one defined in VRRP [10]) at system startup or upon failure of the Master.

A MR periodically announces its presence by sending several parameters over the mobile subnet. For doing so, we have defined a new ICMPv6 message referred to as *Neighbor Router Advertisement* (NRA). We could assume to use existing router advertisements with new options instead, but in addition to their size limitation which restraints the number and the size of new options (ICMPv6 messages can not exceed the minimum IPv6 MTU), this would provide unsolicited information to MNNs. A NRA message includes (but is not necessarily limited to) the ingress interface link-layer address, the ingress interface link-local IPv6 address, the role and the currently used bandwidth ratio of the originator MR coupled with a lifetime. The bandwidth information is a ratio between the bandwidth currently consumed and the overall theoretical bandwidth available on the interface(s) that connects the MR to the Internet. The lifetime refers to the length of time that the provided information is valid. Note that NRAs are sent periodically to the all on-link IPv6 routers multicast address.

The Master MR maintains the status of every Slave MRs in a cache called the Mobile Router Status (MORS) cache. Each MR is registered together with all parameters provided in NRAs. The MORS cache is dynamically updated upon reception of a NRA. Entries from the MORS cache are

automatically deleted once the lifetime that was previously announced in the NRA has expired. Whenever the Master MR has to select a default MR for a MNN, its decision is based on the information currently recorded in the MORS cache.

### B. Default Mobile Router Selection

Our targeted environment provides the mobile subnet with multiple MRs each directly connected to the Internet through its own interface(s). We therefore would like to benefit from these multiple paths to efficiently share the load between all MRs without modifications on the MNNs.

Each MR is statically configured with a virtual link-local IPv6 address which is shared among all MRs. This address is only used for communication between MRs and MNNs. All communications among MRs are achieved with their real and unique link-local IPv6 addresses. Note that only the Master MR defends this virtual address against stateless autoconfiguration performed by another nodes [12]. The rationale behind the use of a virtual link-local IPv6 address is that we can not use a multicast or explicit anycast address as the IPv6 source address of router advertisements has to be a unicast link-local IPv6 address [4]. We could also assume to use the same link-layer address on each MR as suggested in [10], but in our proposal multiple routers are active at the same time. The vision of the same link-layer address on multiples ports simultaneously should be considered as loop by the spanning tree protocol running on layer 2 devices.

In addition, each MR maintains the list of MNNs for which it acts as the default router. As previously mentioned, only the Master MR sends router advertisements over the mobile subnet. In these messages, the Master MR is configured to set the virtual address as the source address and to not set the source link-layer address option. Upon reception of such a message, a MNN could still achieve the IPv6 stateless address autoconfiguration process but no new entries are created yet in the neighbor cache of the MNN (such configuration is defined by the specification of the Neighbor Discovery protocol [4]). As a reminder, the neighbor cache is a set of neighbor's on-link unicast IPv6 address and link-layer address tuples (with additional parameters) to which traffic has been sent recently.

When a MNN wants to send its first data packet to a remote host, it first adds a new entry for its default router in its neighbor cache. The default router is referred to in the neighbor cache by the virtual link-local IPv6 address discovered during the IPv6 autoconfiguration process. At this stage, this entry is in the *incomplete* state which means that the MNN has to resolve the link-layer address of the default router before being able to send a packet to this router. Following the standard procedure, this operation is initiated by sending a Neighbor Solicitation (NS). In our protocol, when a NS is sent to resolve the link-layer address of a router (i.e. the link-layer address associated to the virtual link-local IPv6 address), this message is only intercepted by the Master MR. Upon reception of such a message, the Master MR has to select, among all available MRs, one to which delegate the MNN. This selection could be based on the bandwidth

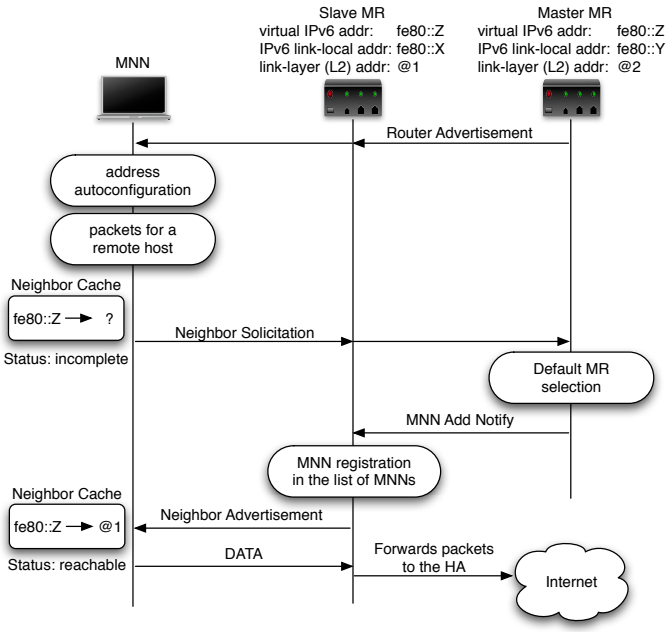


Fig. 1. Default router selection for a MNN

information currently announced by each MR and maintained in the MORS cache. If the selected MR is not the Master itself, we have defined a new message, known as *MNN Add Notify* (MAN), to notify a Slave MR about the delegation of a MNN. This message contains among others the IPv6 address of the targeted MNN. The MR to which the MNN has been delegated then adds the client in its list of MNNs and sends it back a Neighbor Advertisement (NA) with the target address configured to its link-layer address. When the MNN receives this message, it updates its neighbor cache (the entry corresponding to its default router goes to the *reachable* state) and starts to send its data packets to the delegated router. Figure 1 illustrates this procedure.

### C. Failure Detection and Node Redirection

Three cases have to be considered when designing a MR failover mechanism: ingress interface failure, egress interface failure and complete MR failure. The MNNs associated to a MR facing one of these failure cases have to be redirected to a functional MR. Our proposal considers the ingress interface and complete MR failure cases similarly as the failed MR is no longer reachable from the mobile subnet. Furthermore, the mechanisms presented here relies on Neighbor Discovery (the neighbor unreachability detection in particular) to redirect MNNs from one MR to another.

The ingress interface failure of a MR (or complete MR failure) is detected by a neighbor upon non-reception of NRAs whatever the failed MR is configured as Master or Slave. When a certain number of NRAs from the Master MR have been missed by the Slave, it assumes that the Master MR became unavailable. Missing a single NRA does not necessarily imply a Master MR failure as the event may be due to a link layer collision. Upon failure detection, a Slave MR should

initiate the election of a new Master. Note that changing the Master MR while operating the protocol may be expensive as it includes election and potential MNN redirections. As a result, the future election mechanism should ensure that a failed Master coming back online would go to the Slave state. When a Slave MR fails, the corresponding entry in the MORS cache of the Master MR expires. The MNNs bound to a failed MR are automatically redirected to another MR thanks to the neighbor unreachability detection mechanism defined in Neighbor Discovery [4]. Without reachability confirmation, the entry corresponding to the failed MR should be deleted from the neighbor cache of all MNNs bound to the failed MR. This initiates again next-hop determination and address resolution as presented in III-B.

The case of an egress interface failure is managed in a slightly different manner as the failed MR is still able to communicate with the nodes located in the mobile subnet. Our protocol thus also enables explicit reallocations of MNNs from a MR to another. Note that this mechanism can also be used to react upon MR overload (e.g. when the MR attaches to a lower quality access network) and to share the load as fairly as possible (e.g. when a new MR connects to the mobile subnet). The idea lying behind our proposal is to dynamically update the link-layer address of the default router recorded in the neighbor cache of the MNNs.

In order to delegate a set of MNNs to another MR, a MR sends a new ICMPv6 *MNN Redirect* to the Master MR. This message includes the list of MNNs identified together with their link-layer addresses, their link-local IPv6 addresses and their estimated bandwidth requirements. We envision that such selection could be based on preferences such as redirecting first the MNNs which consume most of the bandwidth in order to limit the number of redirected MNNs. Upon reception, the Master MR tries to select a new default MR for each MNN listed in the MNN Redirect message. This selection could be achieved upon matches between the bandwidth requirement of a MNN and the currently available bandwidth on MRs. Once the Master MR has selected a new default MR for a MNN, it sends a MAN message to the selected MR with the *redirect flag* set. Upon reception, the selected MR proceeds with the same actions as presented in III-B. However, a node could only update the cached link-layer address of an existing neighbor cache entry upon reception of a NA with the target link-layer address option and the override flag set [4]. Therefore, the selected MR should set the override flag in every NA transmitted following the reception of a MAN with the redirect flag set. According to [4], a node receiving such a NA has to update the link-layer address in the corresponding neighbor cache entry with the one supplied in the target link-layer address option. Once the Master MR has processed the entire list of MNNs, it sends an ICMPv6 *MNN Redirect ACK* back to the overloaded MR. This message reports the redirection status for each MNN. The redirection status could be set to either *successful* or *unable*. The overloaded MR could therefore remove from its list of MNNs every client for which the redirection status is successful. If the MR is still overloaded, it could try to delegate again several MNNs.

Note that our protocol remains fully compliant with the neighbor unreachability detection procedure [4]. Although all Slave MRs discard multicast NS messages (which are only processed by the Master MR), a MR (either Master or Slave) has to reply to every unicast NS sent to its link-layer address.

## IV. Evaluation of the solution

### A. Implementation Overview

We have implemented the proposed protocol for the GNU/Linux operating system as an userland tool using the Python programming language. This implementation is used in conjunction with NEPL [13], the NEMO BS implementation for the GNU/Linux operating system. Our work uses the Scapy6 packet manipulation program [5] to define the new protocol messages, send them on the medium, and catch that very traffic from the network. Although such an userland implementation might not give as good performance results as a kernel one, it gives interesting preliminary results that we present in the next sections.

### B. The Test Platform

Our test platform is composed of two GNU/Linux routers (representing the MRs), one playing the role of the Master, the other one being the Slave. Those two routers interconnect the same subnetwork to the Internet. This subnetwork represents the mobile network, where two MNNs are located. A correspondent node (CN) is located in the Internet and is used as the communication endpoint of both MNNs. The whole platform is interconnected using Ethernet for the communication medium, as we want to ensure a reliable link to evaluate the behaviour of our protocol

All of those nodes are running the GNU/Linux operating system with a 2.6.22-3 kernel, thus having one of the latest version of the IPv6 stack available on this system. Both NEPL and our implementation are used on the Master and the Slave only. The MNNs do not run any other protocols than the IPv6 protocol suite that is delivered with the system.

### C. Scenario and Results

In all the following scenarios, one of the MR is pre-configured as the Master, the other one taking the role of the Slave. As previously mentioned, router advertisements are only sent by the Master. Results presented here are obtained by taking the most relevant ones among 10 runs of each scenario.

*1) Load sharing among multiple MRs:* In this scenario, two MNNs are located in the mobile subnet. Each MNN wants to send a 300kbps UDP flow to the CN. Each MR connects to the Internet with a 400kbps connectivity. At the beginning, the Master is the only available MR in the mobile subnet, and therefore should become the default MR for the MNNs when they start communicating with the CN. Next, the Slave MR connects to the mobile network and starts sending NRAs in the mobile subnet.

Figure 2 shows the whole experiment on the upper part, and various times around which an event occurs on the lower part.

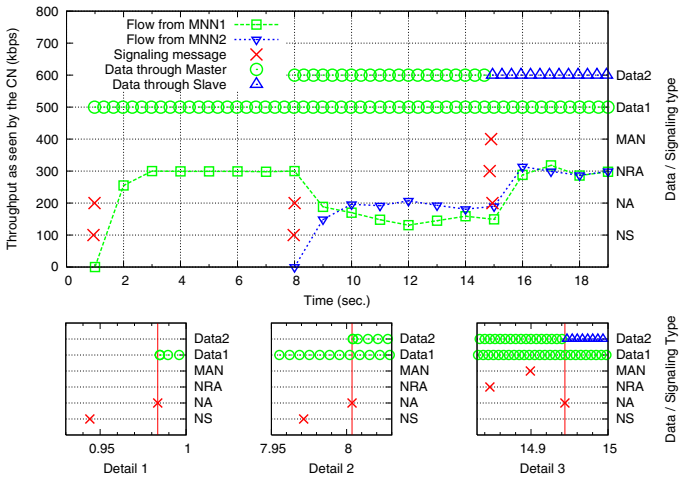


Fig. 2. Load Sharing between Multiple MRs

When the first MNN wants to communicate with the CN, it first tries to resolve the link-layer address of its default router by sending a multicast NS (see Detail 1). Upon reception, the Master adds the MNN to its list of MNNs and replies with a NA. Once the MNN receives this NA, it updates its neighbor cache accordingly and starts sending its data to the CN through the Master. These data packets are represented as *Data1* in the figure (only one packet over 50 is displayed in the upper graph for readability reasons). We can see that the CN starts receiving a flow of 300kbps from the MNN. Later, the second MNN also wishes to initiate a communication with the CN. The same procedure as before applies (see Detail 2) as the Master MR is still the unique MR in the mobile network. When the second MNN also sends its data (represented as *Data2*) through the Master, we can see that the connectivity of the Master can not satisfy simultaneously the needs of both MNNs in terms of bandwidth. The upper graph shows that only 200kbps of each flow is received at the CN which means that each flow experiences a significant degradation. Next, the second MR connects to the mobile network and advertises itself as Slave. Detail 3 shows that upon the reception of the NRA from the Slave, the Master tries to delegate one of the MNN to the Slave by sending a MAN message with the redirect flag set. Upon reception, the Slave accepts the redirection and takes care of updating the neighbor cache of the redirected MNN by sending a NA to it with the override flag set. After the reception of this NA, we can see that all packets from the delegated MNN are sent via the Slave instead of the Master. The MNN has therefore correctly updated its neighbor cache with the link-layer of the Slave. The Master being no longer overloaded, the CN now properly receives the traffic from both MNNs as shown on the figure. This illustrates how our protocol could dynamically distribute the overall bandwidth demand from the MNNs among the available MRs.

*2) Dynamic Redirection of a MNN from a Slave:* In this second experiment, the default router that has been assigned to the MNN is the Slave. The MNN continuously sends an UDP flow to the CN (packets have an average size of 50 bytes and are sent every 10ms). During that communication, the Slave explicitly asks the Master to reassign this MNN to another MR. Such query may happen for example if the Slave detects a failure on its interface connected to the Internet, or if it is overloaded and can no longer guarantee a good quality of service. Note that this scenario differs from the previous one as it illustrates the case of an overloaded Slave MR.

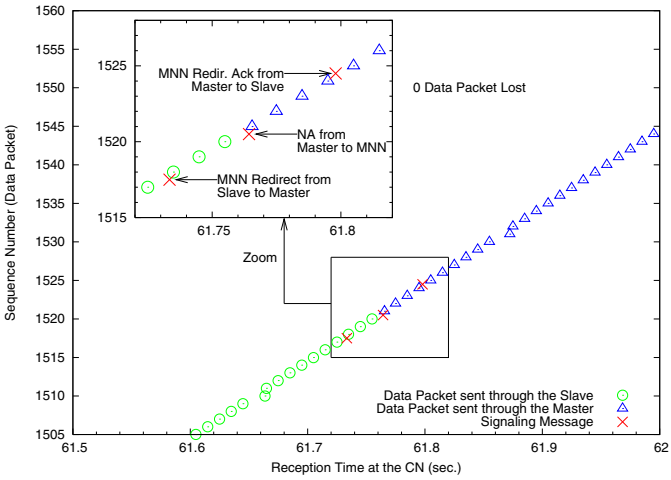


Fig. 3. Dynamic Redirection of a MNN from a Slave

The results are shown in Figure 3. Each dot represents the reception of a packet at the time indicated on the X-axis. The sequence numbers indicated on the Y-axis correspond to the data packet sequence numbers. At t=61.735s, we can see that the Slave MR initiates the redirection of the MNN by sending a MNN Redirect message to the Master. Upon reception, the Master selects itself as the default router for the MNN, then sends a NA (with the override flag set) approximately at t=61.765s to the MNN. From this time, all the packets from the MNN are sent through the Master: the link-layer address of the default router of the MNN has been correctly updated in the MNN neighbor cache. The Master finally sends back to the Slave a MNN Redirect Ack message with a successful status code which completes the redirection procedure. We can underline the fact that no packets have been lost (or even delayed) during the transition. As a result, the default router change remains undetectable for the application on the MNN.

*3) Failover upon router failure:* In this last experiment, the initial configuration and MNN-CN communication pattern is the same as for the second one. This time, the network interface of the Slave connected to the mobile subnet fails without prior notice while the communication is ongoing.

The results are shown in Figure 4. The upper graph shows that the Slave fails at t=13.9s. It is not reachable anymore from the mobile subnet and thus can not either assure the routing of the data packets of the MNN. The states of the entry corresponding to the default router in the MNN neighbor cache reveals that after expiration of the *reachable* state, the MNN starts a neighbor unreachability detection procedure by entering the *delay* state and then the *probe* state. The lower graph of Figure 4 shows that after sending multiple unicast NS

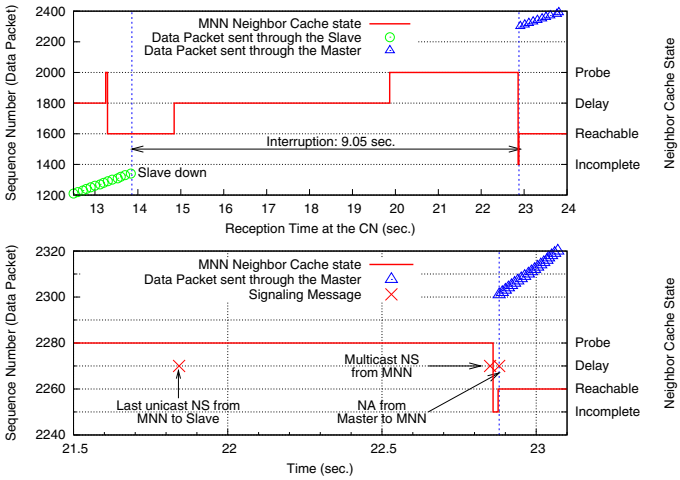


Fig. 4. Failover upon MR failure

to the Slave without receiving any replies, the MNN fallbacks in the *incomplete* state (actually the neighbor cache entry for the default router is deleted but a new one is created right after because the MNN still has data to send). The MNN then immediately tries to resolve the link-layer address of its default router by sending a new multicast NS. In the meantime, the entry corresponding to the Slave in the MORS cache of the Master has been deprecated at t=14.5s. When receiving the multicast NS, the Master assigns the MNN to itself (as it remains the unique MR in the mobile network) and replies a NA to update the neighbor cache of that MNN. Upon reception, the entry on the MNN goes directly to the *reachable* state, which triggers the transmission of the data packets to the CN, through the Master. In order to ensure that the Slave entry will be expired in the MORS cache of the Master before the MNN starts to send a new multicast NS, it is interesting to raise that the lifetime of a MORS entry should not exceed the duration of the *probe* state (fixed to 3 sec. in the Neighbor Discovery specification).

## V. Conclusion

There are various points to take into account when considering the management of multiple mobile routers in the same mobile network. The solutions proposed so far to address the global problem hardly match all the requirements at the same time: they only consider either load sharing or mechanisms to react upon failure. In this paper, we concentrate on the assignment of a default router to the mobile network nodes. Our solution proposes load sharing, dynamic redirection and failover mechanisms in order to fairly use all the resources available in the mobile network. Furthermore, the solution does not require any modifications on the client nodes located in the mobile subnet.

Through preliminary results presented in Section IV, we have validated the behavior of our protocol and confirmed its accuracy with respect to the Neighbor Discovery protocol. Furthermore, explicit redirections of a mobile network node from a mobile router to another remain transparent for ongoing communication. Thanks to the neighbor unreachability detection, the nodes bound to a failed mobile router could automatically retrieve a functional mobile router. However, the default timer values defined for the neighbor unreachability detection may not match latency requirements of time-sensitive communication (we have experienced a flow interruption of approximately 9.05 sec.). We plan to reduce this disruption time by enabling the Master to send unsolicited NA upon failure detection of a Slave instead of changing default timer values. Encouraged by the results obtained for legacy GNU/Linux clients, we also have successfully experimented our protocol with Mac OSX and Windows XP clients.

Our future work in this area is to extend our performance and scalability studies to large scale experiments. In order to reflect bus or train environments, we plan to increase the number of MNNs in the network and consider variable traffic generation models. More error cases will also be studied, such as the failure of the Master which is not evaluated in this document. We are also planning to address the other parts of the NEMO multihoming problem in order to propose a complete solution for NEMO BS. With minor modifications, the home agent would enable multiple mobile routers to register simultaneously the same mobile network prefix. We could also define and exchange routing policies between mobile routers and home agents in order to synchronize upstream and downstream forwarding paths. Especially, we consider to extend [14] for such a purpose.

## References


[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, January 2005.

[2] T. Ernst and K. Uehara, "Connecting Automobiles to the Internet," in *Proc. 3rd International Workshop on ITS Telecommunications (ITST'02)*, Seoul, South Korea, November 2002.

[3] R. Wakikawa, V. Devarapalli, T. Ernst, and K. Nagami, "Multiple Care-of Addresses Registration," IETF Draft, 2008.

[4] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 4861, September 2007.

[5] G. Valadon and A. Ebalard, "Scapy6: IPv6 support for the Scapy interactive packet manipulation program." [Online]. Available: http://hg.natisbad.org/scapy6

[6] C. Ng, T. Ernst, E. Paik, and M. Bagnulo, "Analysis of Multihoming in Network Mobility Support," RFC 4980, October 2007.

[7] S. Cho, J. NA, and C. Kim, "A Dynamic Load Sharing Mechanism in Multihomed Mobile Networks," in *Proc. IEEE International Conference on Communications (ICC'05)*, vol. 3, May 2005, pp. 1459–1463.

[8] N. Choi, J. Ryu, E. Paik, T. Kwon, and Y. Choi, "A Transparent Failover Mechanism for a Mobile Network with Multiple Mobile Routers," *IEEE Commun. Lett.*, vol. 11, no. 7, pp. 604–606, July 2007.

[9] M. Tsukada, T. Ernst, R. Wakikawa, and K. Mitsuya, "Dynamic Management of Multiple Mobile Routers," in *Proc. 13th IEEE International Conference on Networks*, vol. 2, November 2005, pp. 1108–1113.

[10] S. Nadas, "Virtual Router Redundancy Protocol Version 3 for IPv4 and IPv6," IETF Draft, 2008.

[11] T. Ernst, K. Mitsuya, and K. Uehara, "Network Mobility from the InternetCAR perspective," *Journal of Interconnection Networks (JOIN)*, vol. 4, no. 3, September 2003.

[12] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862, September 2007.

[13] "NEPL: a NEMO Basic Support Implementation for UMIP," WIDE Nautilus6 and HUT. [Online]. Available: http://software.nautilus6.org/NEPL-UMIP/

[14] H. Soliman, N. Montavont, N. Fikouras, and K. Kuladinithi, "Flow Bindings in Mobile IPv6 and Nemo Basic Support," IETF Draft, 2007.