A Transparent Failover Mechanism for a Mobile Network with Multiple Mobile Routers

Nakjung Choi, *Student Member, IEEE,* Jiho Ryu, *Student Member, IEEE,* Eunkyoung Paik, Taekyoung Kwon, *Member, IEEE,* and Yanghee Choi, *Senior Member, IEEE*

***Abstract*— In a mobile network that is multihomed by multiple mobile routers, a mobile router that loses link connectivity can be replaced by the other mobile routers. We propose a transparent failover mechanism (TFM) to provide seamless Internet services to nodes in the mobile network, which is validated by implementing a real test-bed. Compared to the network mobility basic support protocol, TFM does not require the nodes attached to the failed mobile router to change their addresses, and hence has two advantages: (a) IP connectivity is maintained transparently, and (b) failover is quickly accomplished by avoiding the address re-configuration process in each node.**

***Index Terms*— Network mobility, multiple mobile routers, multiple CoAs registration, failover.**

I. Introduction

The network mobility working group in IETF has extended the existing mobile IPv6 protocol to support a collective mobility of a mobile network (NEMO) by introducing a mobile router (MR) and a prefix binding update option. A NEMO is viewed and managed as a single unit, which changes its point of attachment to the Internet through the MR's egress interface (EIF). Nodes inside the NEMO are attached to the MR's ingress interface (IIF). These nodes do not need to change their IP addresses as long as they hear router advertisement (RA) messages including the same mobile network prefix (MNP) from the MR, despite the change of the Care-of Address (CoA) of the MR's EIF. Thus, only the MR sends a binding update (BU) message with its own MNP to its home agent (HA), so that the HA can bind the MNP to the new CoA.

Recently, the mobile nodes and multiple interfaces in IPv6 working group in IETF is investigating the multihoming issues for a NEMO but there is no specific mechanism to support multihoming yet. Even though [1] categorizes diverse multihomed NEMO scenarios by the varying numbers of MRs, HAs and MNPs, we believe that the most popular multihoming case will be a NEMO with multiple MRs for redundant Internet connectivity. In such environments, it will be crucial to provide a failover solution among multiple MRs. To this end, we introduce a "*peer*" relationship among the MRs of the same NEMO. For example, suppose that a NEMO has two MRs: MR1 and MR2. We call MR2 is a peer MR (PMR) of MR1 if MR2 can substitute for MR1.

Manuscript received March 13, 2007. The associate editor coordinating the review of this letter and approving it for publication was Christos Douligeris. This research is supported by the Ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21^{st} Century Frontier R&D Program in Korea, and by the Brain Korea 21 project of the Ministry of Education, 2007.

N. Choi, J. Ryu, T. Kwon, and Y. Choi are with the School of Computer Science and Engineering, Seoul National University, Seoul, Korea (e-mail: {fomula, jhryu, tk, yhchoi}@mmlab.snu.ac.kr).

E. Paik is with the Advanced Technology Laboratory, KT, Seoul, Korea (e-mail: euna@kt.co.kr).

Digital Object Identifier 10.1109/LCOMM.2007.070382.

II. Transparent Failover Mechanism

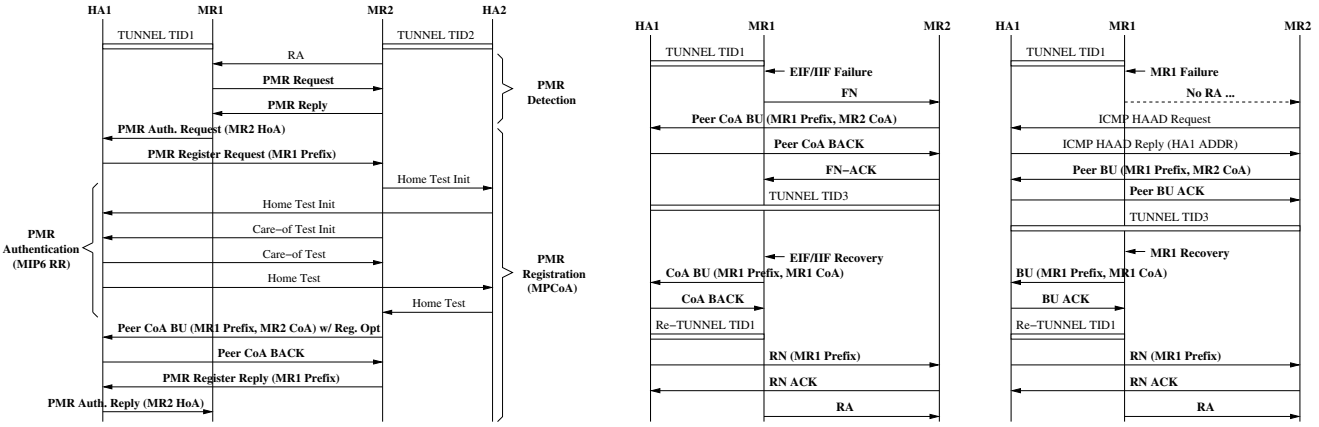
A. Multiple Peer CoA Registration

In TFM, MR2 allowed to provide alternative Internet connectivity to the same NEMO in place of MR1 can be a PMR of MR1. Not only MR1 but also its PMR (MR2) should register their CoAs with the HA of MR1. The CoA of the PMR (Peer CoA) registration is to provide seamless Internet connectivity when MR1 fails or loses link connectivity.

An MR in a NEMO should first authenticate a PMR candidate to prevent malicious nodes from spoofing by using the return routability (RR) procedure of mobile IPv6. Fig. 1(a) illustrates how MR1 authenticates MR2, and triggers MR2 to perform a peer CoA BU. MR1 hearing RA messages from MR2 sends a PMR request message to MR2 in the absence of their peer relationship. MR2 receiving the PMR request message sends a PMR reply message to MR1. Then, MR1 sends a PMR authentication request message including the home address (HoA) of the PMR candidate (MR2) to MR1's home agent (HA1). HA1 then sends a PMR register request message (including the MNP of MR1) to MR2, which triggers a peer CoA BU. The PMR register request message arrives at MR2 via the HA of MR2 (HA2). MR2 receiving the PMR register request message performs the RR procedure to let HA1 authenticate MR2.

After the successful completion of the RR procedure, MR2 sends a peer CoA BU message to inform HA1 of the existence of a PMR (MR2). This message includes the MNP of MR1 and the CoA of the MR2, so that HA1 can register the CoA of MR2 as one of peer CoAs in its binding cache. To distinguish a CoA of an MR that belongs to its HA and peer CoA(s) of PMR(s), the binding cache structure of the HA is to be modified to have a *peer* field which indicates peer CoAs of PMRs. HA1 receiving a peer CoA BU message with a registration option adds a new binding cache entry with the disabled *active*[1] and the enabled *peer* fields corresponding to the MNP1 in the message, and sends back a peer CoA binding acknowledgement (BACK) message. It is noted that a peer CoA BU message updates the entries corresponding to not the HoA of MR1 but the prefix of MR1. After receiving the peer CoA BACK message from HA1, MR2 sends a PMR register reply message including MNP1 in the corresponding

[1]In the modified data structure for a binding cache, the *active* field indicates whether a correspondent tunnel is active or not.



(a) Peer MR registration (b) Egress/Ingress Interface failover (c) MR failover

Fig. 1. Message flows in TFM.

TABLE I
THE CHANGE OF BINDING CACHE AT HA1

HoA	CoA	Active	Peer
MR1 HoA	MR1 CoA	1	0
Prefix	**CoA**	**Active**	**Peer**
MNP1	MR1 CoA	$1 \rightarrow 0 \rightarrow 1$	0
MNP1	MR2 CoA	$0 \rightarrow 1 \rightarrow 0$	1

After a peer CoA BU with and without a registration option.

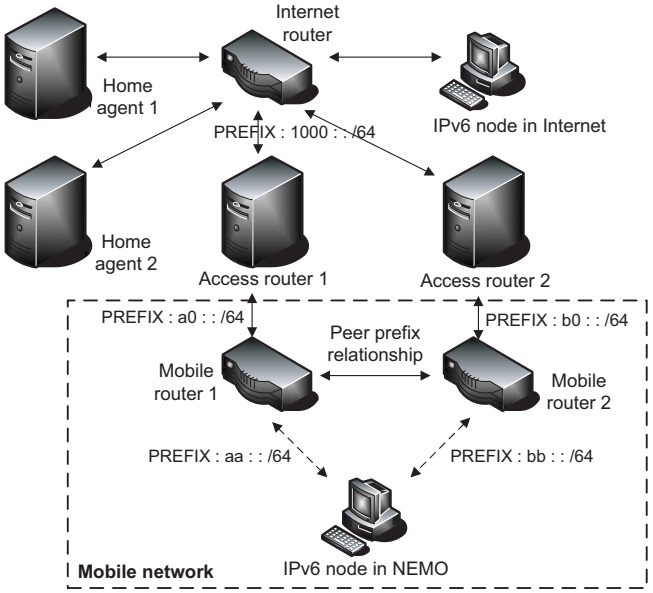


Fig. 2. Test-bed architecture for transparent failover.

PMR register request message to HA1. Finally, HA1 sends a PMR authentication reply message including the CoA of the PMR candidate (MR2) to inform MR1 of whether the PMR registration is successful or not. If the PMR registration is successful, the PMR candidate (i.e., MR2) becomes the PMR of MR1.

B. Failover with Peer CoA BU

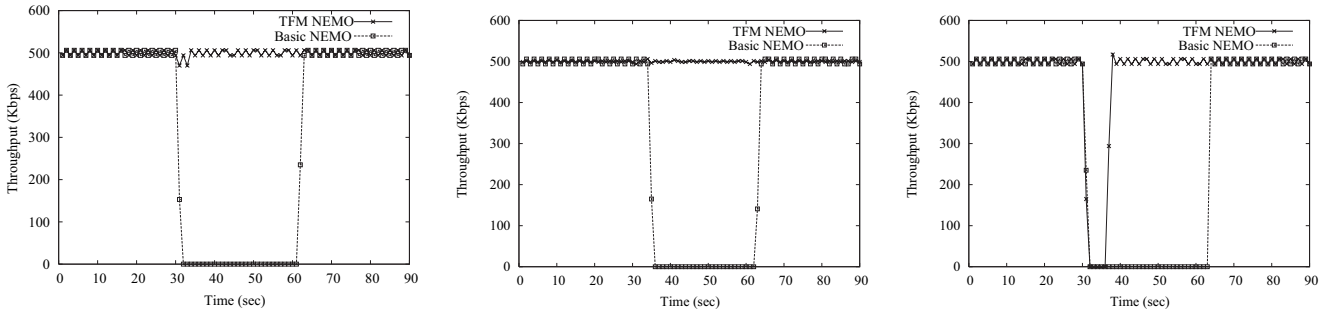
We assume that an MR can detect the failure of its EIF or IIF earlier than any other MRs or nodes, if the MR's EIF/IIF fails, its PMR provides seamless Internet service to nodes in place of the failed MR. Suppose that an EIF (or IIF) of MR1 fails, and MR2 is a PMR of MR1. MR1 sends a failure notification (FN) message including the HA (HA1) address of MR1 through the other interface to inform MR2 of its EIF (or IIF) failure. MR2 receiving this FN message, sends a peer CoA BU message with no registration option to HA1 if MR2 is to take over MR1 considering its policy and load. With the NEMO basic support protocol [2], a peer CoA BU message from other MRs not subordinate to a HA is filtered by the HA. In TFM, a HA receiving a peer CoA BU message with no registration option first looks up its binding cache. In Fig. 1(a), HA1 already has an entry corresponding to MR2 in its binding cache after MR2's successful peer CoA BU with the registration option. HA1 updates the entries in Table I (the left arrows in the $3^r d$ column): HA1 resets the *active* field of the MR1's entry to 0, and sets the *active* field of the MR2's entry to 1.

The HA should always forward incoming packets toward the NEMO through an active tunnel. Therefore, HA1 should set up a HA1-MR2 tunnel[2] after MR2 successfully completes the peer CoA BU without the registration option. Then, MR2 sends additional RA messages containing MNP1 of MR1 to the NEMO instead of MR1, so that the nodes that have been connected to MR1 can now use MR2 without IP address changes. They preserve their sessions by using the HA1-MR2 tunnel that bypasses ingress filtering. Additionally, MR2 should forward outgoing packets from the nodes in the NEMO through an appropriate tunnel (between MR2-HA2 and MR2-HA tunnels) based on the prefix of the source address of the packets. When the failed interface of MR1 is repaired, MR1 sends a CoA BU message to HA1. HA1 receiving this CoA BU message resets the *active* field of the MR2's entry to 0, which corresponds to the prefix in the CoA BU message. Also, HA1 sets the *active* field of the MR1's entry to 1 as shown in Table I (the right arrows in the $3^r d$ column).

When MR1 itself fails, MR1 cannot send the FN message, so that its PMRs know neither the MR1's failure nor the HA1 address of MR1. In this case, if MR2 does not hear a predetermined number[3] of consecutive RA messages from

[2]A bi-directional tunnel between a HA and a PMR can be in advance established for fast failover after the PMR registration.

[3]In our experiments, this threshold is set to 3.



(a) EIF failure & recovery (b) IIF failure & recovery (c) MR failure & recovery

Fig. 3. Throughput over time (RA interval = random[0.5s, 1.0s]).

MR1, MR2 concludes that MR1 itself failed. Therefore, MR2 can first perform a dynamic home agent address discovery (DHAAD) in mobile IPv6 to obtain the HA1 address of MR1. After finishing DHAAD, MR2 obtains the HA1 address and sends the peer CoA BU message with no registration option to HA1, and hence sets up a tunnel. The recovery process of the failed MR is the same as that of an EIF (or IIF) failure.

III. Implementation and Evaluation

A. TFM NEMO Test-Bed

We first built a basic NEMO test-bed with NEPL (NEMO Platform for Linux) [3] which is a well-known open source for Linux. As shown in Fig. 2, the test-bed consists of 9 Linux machines: two IPv6 nodes, two MRs, two HAs, one Internet router, and two access routers (ARs). Each machine in our test-bed uses Fedora Core 4 on Intel Pentium 4 2.8Ghz CPU and 512MB RAM. Especially, each MR has two ethernet ports for egress and ingress interfaces, respectively. We use Dynamic Switch [4] (skipped in Fig. 2) to emulate the dynamic connectivities on wireless links among the IPv6 node in the NEMO, the MRs in the NEMO, and the ARs.

Then, we implemented TFM, which consists of two daemons: *failoverd* and *tunneld* on the test-bed. The role of *failoverd* in an MR is to monitor its ingress and egress interfaces and the aliveness of the corresponding PMRs. If *failoverd* detects PMR candidates by hearing RA messages with other MNPs, it performs a PMR authentication and registration. Thereafter, if *failoverd* detects the failure of an ingress or egress interface, it terminates *radvd* which is sending RA messages and sends a FN message to the MR's PMR. The *failoverd* of an MR which receives the FN message or does not hear RAs from its PMR triggers new *radvd* to advertise the MNP of the failed MR and then invokes *tunneld*. As *nemod* in NEPL sets up and tears down a bi-directional tunnel between an MR and its HA, so *tunneld* in TFM establishes/tears down a bi-directional tunnel between the PMR and the HA of the failed MR.

B. Evaluation

We compare the NEMO basic support protocol and TFM by measuring the throughput of an IPv6 node belonging to the failed MR over time. A traffic generating tool called Iperf [5] is used to download IPv6 UDP traffic from a correspondent IPv6 node in Internet transmitting at the data rate of 500Kbps. During the total experiment time of 90s, we intentionally make one of three failures (i.e., ingress interface, egress interface, and MR failure) at 30s at MR1 and the failure recovers at 60s. As shown in Fig. 3, the NEMO basic support protocol cannot provide a continuous Internet service to the IPv6 node belonging to the failed MR (MR1) whichever failure occurs. Even if the IPv6 node performs an IPv6 re-configuration process, the original UDP session cannot be sustained. However, in TFM, the IPv6 node can use an almost seamless Internet service despite any of three failures because the PMR (MR2) of the failed MR (MR1) provides an alternative Internet service almost immediately after detecting the failure. In Fig. 3(c), even with TFM, the throughput of the IPv6 node is zero for a few seconds. This is because it takes time for MR2 to detect MR1's failure due to the absence of a FN message. Even though IPv6 re-configuration is applied, TFM can restore the original throughput much earlier because the combination of router failure detection and global IPv6 address auto-configuration takes a long time [6].

IV. Conclusion

We propose a transparent failover mechanism (TFM) for a NEMO, multihomed by multiple MRs, by using a *peer* relationship. In our proposal, PMRs can provide a seamless Internet service to nodes that have been attached to a given MR when the MR fails due to the movement or the wireless link disconnection. TFM provides transparent failover in the sense that nodes do not need to change their IP addresses. Experimental studies reveal that TFM has a negligible disruption except for system failure. Since IP handover of an MR may take a long time depending on the RA intervals from ARs, TFM can be leveraged for seamless Internet services, which means that a PMR can provide an interim connectivity while an MR performs handover.

References

[1] C. Ng, T. Ernst, E. Paik, and M. Bagnulo, "Analysis of multihoming in network mobility support," Internet draft draft-ietf-nemo-multihoming-issues-07, Feb. 2007.
[2] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol," RFC 3963, Jan. 2005.
[3] NEMO Platform for Linux, http://software.nautilus6.org/NEPL/.
[4] Linux Dynamic Switch, http://sourceforge.net/projects/dynamic-switch/.
[5] Iperf, http://dast.nlanr.net/Projects/Iperf/.
[6] C. Vogt, R. Bless, M. Doll, and G. Daley, "Analysis of IPv6 relocation delays," Internet Draft draft-vogt-dna-relocation-01.txt, July 2005.