NEMO Working Group
Internet-Draft
Expires: April 11, 2005

P. Thubert
Cisco
N. Montavont
LSIIT - ULP

October 11, 2004

Nested Nemo Tree Discovery
draft-thubert-tree-discovery-01.txt

Status of this Memo

Copyright Notice

Abstract

The purpose of this paper is to describe a minimum set of features that extends the Nemo basic support in order to avoid loops in the nested Nemo case. As a result, Mobile Routers assemble into a tree that can be optimized based on various metrics.

Table of Contents

1. Introduction

As per Nemo Basic support, a Mobile Router autoconfigures a single Care of Address to register to its Home Agent and terminate its MR-HA tunnel. That CoA is the MR's point of attachment to the nested Nemo.

Consequently, if loops are avoided, the nested Nemo assumes the shape of a tree. The nodes of the tree are Mobile Routers, the root is either a fixed or a Mobile Router, called in the latter case the root Mobile Router in NEMO terminology [6]. The leaves are mobile or fixed hosts, called Local Fixed Nodes, Local Mobile Nodes and Visiting Mobile Nodes in the NEMO terminology.

This paper provides (1) a minimum extension to IPv6 Neighbour Discovery Router Advertisements in order to ensure that Mobile Routers attaching to one another actually avoid loops and end up forming a tree, and (2) the minimum common part of all MR algorithms that is required to ensure that whatever their specific decisions, loops between MRs will be avoided.

The method is based on an autonomous decision by each MR with no global state convergence such as a MANET proactive routing protocol. In fact, MRs may make different decisions from a same input, based on their own configuration and their own algorithms.

In order to build trees of Mobile Routers, we propose an extension to the ICMP Router Advertisement (RA) message, the Tree Information Option (TIO). The RA/TIO allows MRs to advertise the tree they belong to, and to select and move to the best location within the available trees. MRs propagate the TIO down the tree, updating some metrics such as the tree depth, leaving alone root information such as the tree identifier, and sending the result in RAs over the ingress interfaces.

2. Terms and Abbreviations

This document assumes that the reader is familiar with Mobile IPv6 as defined in [3] and with the concept of Mobile Router defined in the Nemo terminology document [6].

For the needs of this paper, the following new definitions are introduced:

Nemo Clusterhead: The root of a tree of mobile routers. When the tree of Mobile Routers is attached to the infrastructure, the fixed Access Router may act as cluster head if it supports the Tree Information Option described in this document. If it does not, then the Clusterhead coincides with the root MR in NEMO

terminology. A clusterhead is elected even when the tree is not attached to the infrastructuture. A stand-alone MR is a Clusterhead.

Floating Tree: A Nested Nemo which Clusterhead is a MR that is not attached to an AR.

Grounded Tree: A Nested Nemo whose Clusterhead is attached to the infrastructure. In other words, the clusterhead is either a fixed router that supports RA/TIO or is a MR which attachment router is a fixed router that does not support RA/TIO.

Mobile Access Router: A Mobile Router that provides Access Router services to other MRs.

Attachment Router: The Router that is selected as Access Router by a Mobile Router, making it its parent in the nested NEMO tree.

Propagation: The action by a Mobile Router that consists in receiving a RA/TIO from its Attachment Router, recomputing a few specific fields, removing unknown suboption, and appending the resulting TIO to RAs sent over the ingress interfaces.

3. Motivations

3.1 Multihomed nested mobile network

A nested mobile network that is made of multiple MRs having a direct connection to the Internet is said to be multihomed. Multihoming in Nemo offers useful properties to MNNs. The NEMO multihoming issues [8] draft lists potential multihomed configurations for Nemo and explains the different problems and advantages that some configurations may introduce. Multihoming offers three main abilities to the Nemo: it allows route recovery on failure, redundancy and load-sharing between MRs (or between MRs'interfaces). However, for the moment there are no requirements nor protocol defining how to use several egress interfaces inside a Nemo.

In a nested Nemo, the hierarchy of MRs increases the complexity of the route and/or router selection for MNNs. Each level of a Nemo implies the usage of a new tunnel between the MR and its home agent. Thus if a MNN connects to a sub-Nemo which is also a sub-Nemo, packets from the MNN will be encapsulated three times.

When the Nemo where the MN is connected to is multihomed, the MN may have the choice between several AR to be its default router. Reference [9] introduces new options in Router Advertisement to allow

any node on a link to choose between several routers. This option mainly consists of a 2-bits flag that indicates the preference of the router (low, medium or high). Furthermore, the same flag can be set in the Route Information option indicating the preference of a specific prefix. Therefore, any node can determine its best default router(s) according to a given destination and its best router for default, which will be used by default.

However this preference is only useful in a flat topology; It gives a way to the node to choose between different access routers advertising prefixes on the node link. But if the node is inside a hierarchical topology (some access routers are not at the same level) the node can not learn the level of each access router.

One of the usage of the new option introduced in this document is to distribute information on the hierarchy of MRs. This information can be distributed to ARs, MRs and MNNs as well in order to allow better route selection and to increase the knowledge of the Nemo topology on each node.

3.2 Loops in nested Nemo

When several MRs attach to each other to form a nested Nemo, loops can be created if they are not explicitely avoided. In the simplest case, when egress and ingress interfaces of an MR are all wireless, a mobile router may be listening to Router Advertisement from its own ingress interface, creating a confliction problem. In the general case, arbitrary attachment of MRs will form graphs that are not exempt of loops. For instance: Assume a nested Nemo where MR1 is connected to the infrastructure, and MR3 is attached to MR2. Say that MR2 can hear both MR3 and MR1 over its wireless egress interface. If MR2 select MR1, the connectivity to the infrastruture is provided for all. But if MR2 selects MR3, MR2 and MR3 end up forming a loop and are disconnected from their Home Agents.

With Nemo basic support, a MR uses a single primary CareOf to attach to the nested structure. As a result, if loops are avoided, the nested NEMO end up forming a tree. It is beneficial to be able to form that tree in an optimum fashion for a given set of metrics such as tree depth.

The shape of a nested Nemo may change rapidly due to MRs movement. It is thus impractical to expect each MR to be able to maintain states about the whole tree structure in a link state fashion. On the contrary, it is also beneficial to allow each MR to make its own independent selection based on a minimum information about its immediate neighbors, in order to reestablish the tree quickly upon erratic movements.

Each MR should be able to make its own attachment router selection based on its own condition (eg battery level), its own set of constraints that may not apply to other MRs in the tree, and in general its own algorithm. As a result, the standardization effort should concentrate on a common minimum set of rules that must be common to all MRs in order to prevent routing loops in the nested NEMO while leaving MRs independent in their AR selection algorithms.

4. Router Advertisement extensions

New extensions of Router Advertisement are proposed to distribute the knowledge of the MR hierarchy inside a nested Nemo. These extensions are defined in different options/sub-options: a flag bit from the reserved flag field of Router Advertisement message is used to

indicate whether the sending router is a MR or not; a new option is defined to transport minimum information on the tree to avoid loops generation;

4.1 Router Advertisement message

We propose to use a reserved flag of the Router Advertisement message to inform whether the sending router is a MR or not.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |     Code      |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Cur Hop Limit |M|O|H|N|Reservd|       Router Lifetime         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Reachable Time                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          Retrans Timer                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Options ...
    +-+-+-+-+-+-+-+-+-+-+-+-
```

Nemo enabled router (N)

The Nemo enabled router (N) bit is set when the sending router is a Mobile Router.

4.2 Tree Information Option

The following option regroups the minimum information that allows a MR to discover a tree and select its point of attachment while avoiding loop generation. It can also be used by MNNs to select their best default router. If the default router of a non-MR sends

Router Advertisements with a tree discovery option, the non-MR MUST set the N flag of its own Router Advertisement to 0 and copy the Tree Discovery Option in its own Router Advertisement.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|     Type      |    Length     |G|H|       Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Preference   |               BootTimeRandom                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TreeDepth   |   TreePref.   |         TreeDelay             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PathDigest                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                            TreeID                             |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   sub-option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. Tree Information Option

Type 8-bit unsigned integer set to 10 by the Clusterhead.

Length 8-bit unsigned integer set to 4. The length of the option (including the type and length fields) in units of 8 octets.

Grounded (G) The Grounded (G) flag is set when the Clusterhead is attached to a fixed network infrastructure (such as the Internet).

Home (H) The Home (H) flag is set when the Clusterhead is attached to its home network.

Reserved 16-bit unsigned integer set to 0 by the Clusterhead.

Preference The administrative preference of that (mobile) Access Router. Default is 0. 255 is the highest possible preference. Set by each MR at propagation time.

BootTimeRandom A random value computed at boot time and recomputed in case of a duplication with another AR. The concatenation of the Preference and the BootTimeRandom is a 32-bit extended preference that is used to resolve collisions. It is set by each MR at propagation time.

TreeDepth 8-bit unsigned integer. The tree depth of the clusterhead is 0 if it is a fixed router and 1 if it is a Mobile Router. The tree Depth of a tree Node is the depth of its attachment router as received in a TIO, incremented by one. All nodes in the tree advertise their tree depth in the Tree Information Options that they append to the RA messages over their ingress interfaces as part of the propagation process.

TreePreference 8-bit unsigned integer set by the Clusterhead to its preference and unchanged at propagation. Default is 0 (lowest preference).

TreeDelay 16-bit unsigned integer set by the Clusterhead indicating the delay before changing the tree configuration, in milliseconds. A default value is 128ms. It is expected to be an order of magnitude smaller than the RA-interval so if the clusterhead has a sub-second RA-interval, the Tree delay may be shorter than 100ms. It is also expected to be an order of magnitude longer than the typical propagation delay inside the nested Nemo.

PathDigest 32-bit unsigned integer CRC, updated by each MR. This is the result of a CRC-32c computation on a bit string obtained by appending the received value and the MR Care of Address. Clusterheads use a 'previous value' of zeroes to initially set the PathDigest.

TreeID 128-bit unsigned integer which uniquely identify a tree set by the Clusterhead. The global IPv6 home address of the Clusterhead can be used.

5. Tree Discovery

Here follows a set of rules and definitions that MUST be followed be all Mobile Routers:

1. An MR that is not attached to an AR is the Nemo Clusterhead of its own, floating tree.

2. An MR that is attached to an AR that does not support TIO, is the clusterhead of its own, grounded tree.

3. A router sending a RA without TIO is considered a grounded AR at depth 0. Thus, a MR that attaches to an AR that does not include a TIO in its RA becomes a Clusterhead of depth 1.

4. The Nemo ClusterHead of a tree exposes the tree in the RA/TIO and MRs propagate the TIO down the tree with the RAs that they forward over their ingress links.

5. An MR that is already part of a tree MAY move at any time and with no delay in order to get closer to the clusterhead of its current tree - i.e. in order to reduce its own tree depth. But an MR MUST NOT move down the tree that it is attached to. MRs MUST ignore RAs that are received from other routers located deeper or at the same depth within the same tree.

6. An MR may move from its current tree into any different tree at any time and whatever the depth its reaches in the new tree, but it may have to wait for a Tree Hop timer to elapse in order to do so. The MR will join that other tree if it is more preferable for reasons of connectivity, configured preference, size, security, bandwidth, tree depth, or whatever metrics the MR cares to use.

7. If a MR has selected a new attachment router but has not moved yet (because it is waiting for Tree Hop timer to elapse), the MR is unstable and refrains from sending RATIOs.

8. When an MR joins a tree, moves within its tree, or when it receives a modified TIO from its current access router, the MR sends an unsolicited Router Advertisement message on all its mobile networks (i.e. all its ingress interfaces). The RA contains a TIO that propagates the new tree information. At the same time, the MR MAY send a Binding Update to its home agent or a local proxy of some sort, because the tree it is attached to has changed. If the MR fails to reach its HA, it MAY attempt to roll back the movement or to retry the HA discovery procedure.

9. This allows the new higher parts of the tree to take place first eventually dragging their sub-tree with them, and allowing stepped sub-tree reconfigurations, limiting relative movements.

5.1 tree selection

The tree selection is implementation and algorithm dependent. In order to limit erratic movements, and all metrics being equal, MRs SHOULD stick to their previous selection. Also, MRs SHOULD provide a mean to filter out candidate ARs whose availability is detected as

fluctuating, at least when more stable choices are available. For instance, the MR MAY place the failed AR in a Hold Down mode that ensures that the AR will not be reused for a given period of time.

The known trees are associated with the AR that advertises them and kept in an ordered list, per order of preference, by extending the Default Router List. DRL entries are extended to store the information received from the last TIO, and a timer ID -and related data- to delay the reaction to a better RA message, the tree hop timer, as described in the next section.

When connection to a fixed network is not possible or preferable for security or other reasons, scattered trees should aggregate as much as possible into larger trees in order to allow inner connectivity. How to balance these trees is implementation dependent, and MAY use a specific visitor-counter suboption in the TIO.

5.2 Subtree mobility

It might be perceived as beneficial for a subtree to move as a whole. The way it would work is for a MR to stay root-MR even if itself is attached into a parent tree. But the loop avoidance is based on the knowledge of the tree that the MR visit, preventing a MR to move down a same tree. So without additional support, tree-level loops could form.

To avoid this, it is possible to add a path vector suboption to the TIO that reflects the nesting of trees. If a root-MR joins a parent tree, then it needs to add its treeID to the path vector, but it can not join if the treeID is already listed.

A specific case is the root-MR of a tree that attaches to a fixed Access Router. That root-MR might omit to consider a TIO that comes from the new AR and decide to stay root, in order to keep the tree consistency from the nested MRs standpoint. This does not create loops, even if the path vector is not present

5.3 Tree Hop Timer

The tree Hop timer serves 2 purposes:

Delay the reattachment of a subtree that has been forced to detach. This allows to make sure that when a subtree has detached, the RATIO that is initiated by the new clusterhead has spread down the subtree so that two different trees have formed.

Limit RA/TIO storms when two trees collide. The idea is that between the nodes from tree A that wish to move to tree B, those

that see the highest place in tree B will move first and advertise their new locations before other nodes from tree A actually move.

A new tree is discovered upon a router advertisement message with or without a RA/TIO. The MR joins the tree by selecting the source of the RA message as its access router (default gateway) and propagating the TIO accordingly.

When a new tree is discovered, the candidate AR that advertises the new tree is placed in a held up state for the duration of a Tree Hop timer. If the new AR is more preferable than the current one, the MR expects to jump and becomes unstable.

A MR that is unstable may discover other ARs from the same new tree during the instability phase. It needs to start a new Tree Hop timer for all these. The first timer that elapses for a given new tree clears them all for that tree, allowing the MR to jump to the highest position available in the new tree.

The duration of the Tree Hop timer depends on the tree delay of the new tree and on the depth of AR that triggers it:

(AR's depth + random) * AR's tree_delay (where 0 <= random < 1). It is randomized in order to limit collisions and synchronizations.

5.4 Collisions and stability

Attachment routers in the DRL may or may not be usable for roaming depending on runtime conditions. The following states are defined:

Current This AR is used for roaming

Candidate This AR can be used for roaming. Typically, an initial held-up period is over but the candidate is not preferred over the current AR.

Held-Up This AR can not be used till tree hop timer elapses. This does not occur for a fixed AR that does not send a TIO since the tree delay is null in that case..

Held-Down This AR can not be used till hold down timer elapses. At the end of the hold-down period, the router is removed from the DRL, and will be reinserted if it appears again with a RA.

Collision This AR can not be used till its next Router Advertisement

5.4.1 Hold down

When a router is 'removed' from the Default Router List, it is actually held down for a hold down timer period, in order to prevent flapping. This happens when an AR disappears (upon expiration timer), and when an AR is tried but can not reach the HA (upon expiration of another AR, or upon tree hop for that AR).

An Attachment Router that is held down is not considered for the purpose of roaming. When the hold down timer elapses, the AR is removed from the DRL.

5.4.2 Instability

A MR is instable when a Held-Up AR is placed before the current AR. Instability is transient (In the order of tree hop timers). When a MR is instable, it MUST NOT send RAs with TIO. This avoids loops when MR A wishes to attach to MR B and MR B wishes to attach to MR A. Unless RA cross (which is a short window of time), a MR receives TIO from stable ARs, which do not plan to attach to itself, so the MR can safely attach to them.

A MR is instable when it is prepared to move shortly to another AR. This happens typically when it discovers a new and more preferred candidate AR, for the duration of the tree hop timer. During that time, the new candidate is placed in Held-up state. Instability may also occur when the current AR is lost and the next best is still held up. Instability is resolved when the tree hop timer of all the AR (s) causing instability elapse. Such AR is changes state to Current or Held-Down.

5.4.3 Collision

A race condition occurs if 2 MRs send RA/TIO at the same time and wish to join each other. In order to detect the situation, MRs time stamp the sending of RA/TIO. Any RA/TIO received within a short media-dependant period introduces a risk. To divide the risk, A 32bits extended preference is added in the TIO. The first byte is the AR own preference (default 0), the rest is a boot time computed random.

A MR that decides to join an AR will do so between (AR depth) and (AR depth + 1) times the AR tree delay. But since a MR is unstable as soon as it receives the RA/TIO from the preferred AR, it will restrain from sending a RA/TIO between the time it receives the RA and the time it actually jumps. So the crossing of RA may only happen during the propagation time between the AR and the MR, plus some internal queuing and processing time within each machine. It is

expected that one tree delay normally covers that interval, but ultimately it is up to the implementation and the configuration of the AR to define the duration of risk window.

There is risk of a collision when a MR receives an RA, for an other mobile router that is more preferable than the current AR, within the risk window. In the face of a potential collision, the Mobile Router with the lowest extended preference processes the RA/TIO normally, while the router with the highest preference places the other in collision state, does not start the tree hop timer, and does not become instable. It is expected that next RAs between the two will not cross anyway.

5.5 Legacy Routers

A legacy router sends its Router Advertisements without a TIO. Consequently, a legacy router can be mistaken for a fixed Access Router when it is placed within a nested NEMO structure, and defeat the loop avoidance mechanism. Consequently, it is important for the administrator to prevent address autoconfiguration by visiting MRs from such a legacy router.

6. Changes

6.1 Changes from version 00 to 01

Added text on subtree mobility from the discussion with Marcelo.

Added text on nested legacy routers from the discussion with Marcelo.

7. Acknowledgments

The authors wish to thank Marco Molteni and Patrick Wetterwald (cisco) for their participation to this design and the review of the document, and Massimo Villari (university of Messina), for his early work on simulation and research on the subject. This work is also based on prior publications, in particular HMRA [7] by Hosik Cho and Eun-Kyoung Paik from Seoul National University and other non IETF publications coauthored with Thierry Ernst and Thomas Noel. Finally, thanks to Marcelo Bagnulo Braun for his constructive review.

8 References

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

[2] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[3] Johnson, D., Perkins, C. and J. Arkko, "Mobility Support in IPv6", draft-ietf-mobileip-ipv6-24 (work in progress), July 2003.

[4] Devarapalli, V., "Network Mobility (NEMO) Basic Support Protocol", draft-ietf-nemo-basic-support-03 (work in progress), June 2004.

[5] Ernst, T., "Network Mobility Support Goals and Requirements", draft-ietf-nemo-requirements-02 (work in progress), February 2004.

[6] Ernst, T. and H. Lach, "Network Mobility Support Terminology", draft-ietf-nemo-terminology-01 (work in progress), February 2004.

[7] Cho, H., "Hierarchical Mobile Router Advertisement for nested mobile networks", draft-cho-nemo-hmra-00 (work in progress), January 2004.

[8] Ernst, T., "Analysis of Multihoming in Network Mobility Support", draft-ietf-nemo-multihoming-issues-00 (work in progress), July 2004.

[9] Draves, R. and R. Hinden, "Default Router Preferences and More-Specific Routes", draft-ietf-ipv6-router-selection-03 (work in progress), December 2003.

Authors' Addresses

Pascal Thubert
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 4 97 23 26 34
EMail: pthubert@cisco.com

Nicolas Montavont
LSIIT - Univerity Louis Pasteur
Pole API, bureau C444
Boulevard Sebastien Brant
Illkirch 67400
FRANCE

Phone: (33) 3 90 24 45 87
EMail: montavont@dpt-info.u-strasbg.fr
URI: http://www-r2.u-strasbg.fr/~montavont/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.