Nested Nemo Tree Discovery
draft-thubert-tree-discovery-08.txt

Status of this Memo

Copyright Notice

Abstract

This paper describes a simple distance vector protocol that exposes only a default route towards the infrastructure in a nested NEMO configuration. The draft extends the Neighbor Discovery Protocol (Narten, T., Nordmark, E., Simpson, W., and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” September 2007.) [RFC4861] in order to carry information and metrics which will help a Mobile Router select its Attachment Router(s) in an autonomous fashion and provides generic rules which guarantee that the interaction of different selection processes will not create loops.

Table of Contents

---

1. Introduction

TOC

As per Nemo Basic support (Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, “Network Mobility (NEMO) Basic Support Protocol,” January 2005.) [RFC3963], a Mobile Router autoconfigures a single Care of Address (CoA) to register to its Home Agent and terminate its Mobile Router-Home Agent tunnel. That Care of Address is the Mobile Router 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 (Ernst, T. and H. Lach, “Network Mobility Support Terminology,” November 2006.) [I-D.ietf-nemo-terminology]. 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 Neighbor 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 Mobile Router algorithms that is required to ensure that whatever their specific decisions, loops between Mobile Routers will be avoided.
The method is based on an autonomous decision by each Mobile Router with no global state convergence such as a MANET proactive routing protocol. In fact, Mobile Routers 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 TIO allows Mobile Routers to advertise the tree they belong to, and to select and move to the best location within the available trees. Mobile Routers propagate the TIO in RA 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

TOC

This document assumes that the reader is familiar with Mobile IPv6 as defined in [RFC3775] (Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," June 2004.) and with the concept of Mobile Router defined in the Nemo terminology document [I-D.ietf-nemo-terminology] (Ernst, T. and H. Lach, "Network Mobility Support Terminology," November 2006.).
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 Mobile Router in NEMO terminology. A clusterhead is elected even when the tree is not attached to the infrastructure. A stand-alone Mobile Router is a clusterhead.

Floating Tree: A Nested Nemo which clusterhead is a Mobile Router that is not attached to an Access Router.

Grounded Tree: A Nested Nemo whose clusterhead is attached to the infrastructure. In other words, the clusterhead is either a fixed router that supports Router Advertisement - Tree Information Option or is a Mobile Router which attachment router is a fixed router that does not support Router Advertisement - Tree Information Option.

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

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 Router Advertisement Tree Information Option from its Attachment Router, recomputing a few specific fields, removing unknown suboptions, and appending the resulting TIO to RAs sent over the ingress interfaces.

Uniform Path Metric: A multihop metric that can be used by Tree Discovery plug-ins to select feasible successors and specifically an Attachment Router.

3. Motivations

TOC

3.1. Multi-Homed nested mobile network

TOC

A nested mobile network that is made of multiple Mobile Routers having a direct connection to the Internet is said to be multi-homed. Multihoming in Nemo offers useful properties to Mobile Network Nodes. The NEMO multihoming issues (Ng, C., “Analysis of Multihoming in Network Mobility Support,” February 2007.) [I-D.ietf-nemo-multihoming-issues] draft lists potential multi-homed 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 Mobile Routers (or between interfaces of a given Mobile Router). However, for the moment, there is no requirements nor protocol that would define in interaction between several egress interfaces inside a Nemo.
In a nested Nemo, the hierarchy of Mobile Routers increases the complexity of the route and/or router selection for Mobile Network Nodes. Each level of a Nemo implies the usage of a new tunnel between the Mobile Router and its home agent. Thus if a Mobile Network Node connects to a sub-Nemo which is also a sub-Nemo, packets from the Mobile Network Node will be encapsulated three times.
When the Nemo where the MN is connected to is multi-homed, the MN may have the choice between several Attachment Router to be its default router. Reference [RFC4191] (Draves, R. and D. Thaler, “Default Router Preferences and More-Specific Routes,” November 2005.) 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 attachment routers advertising prefixes on the node link. But if the node is inside a hierarchical topology the node can not learn the depth of each attachment router, and might not select the most efficient path. In particular, there is a need for Uniform Path Metric that accounts for a multihop wireless path.

One of the usage of the new option introduced in this document is to distribute information on the hierarchy of Mobile Routers. This information can be distributed to Attachment Routers, Mobile Routers and Mobile Network Nodes 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

TOC

When several Mobile Routers attach to each other to form a nested Nemo, loops can be created if they are not explicitly avoided. In the simplest case, when egress and ingress interfaces of A Mobile Router 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 Mobile Routers will form graphs that are not exempt of loops. For instance: Assume a nested Nemo where Mobile Router1 is connected to the infrastructure, and Mobile Router3 is attached to Mobile Router2. Say that Mobile Router2 can hear both Mobile Router3 and Mobile Router1 over its wireless egress interface. If Mobile Router2 select Mobile Router1, the connectivity to the infrastructure is provided for all. But if Mobile Router2 selects Mobile Router3, Mobile Router2 and Mobile Router3 end up forming a loop and are disconnected from their Home Agents.

With Nemo basic support, a Mobile Router uses a single primary Care Of Address 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 Mobile Routers movement. It is thus impractical to expect each Mobile Router 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 Mobile Router 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 Mobile Router 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 Mobile Routers 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 Mobile Routers in order to prevent routing loops in the nested NEMO while leaving Mobile Routers independent in their Attachment Router selection algorithms.

---

4. Tree Information Option

TOC

The Tree Information Option carries a number of metrics and other information that allows a Mobile Router to discover a tree and select its point of attachment while avoiding loop generation.

4.1. TIO base option

TOC

The Tree Information Option is a container option, which might contain a number of suboptions. The base option regroups the minimum information set that is mandatory in all cases.

```
 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|B| Reserved|  Sequence     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TreePref.    |        BootTimeRandom                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MR Preference |   TreeDepth   |         TreeDelay             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           PathDigest                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                            TreeID                             |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   sub-option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: TIO base option

Type: 8-bit unsigned integer set to 10 by the clusterhead. Value is "TBD".

Length: 8-bit unsigned integer set to 4 when there is no suboption. The length of the option (including the type and length fields and the suboptions) 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 Mobile Router is bound to its home network over NEMO. With NEMO Basic Support,a MR that is not bound Home cuts off its visitors from the Internet as well. This can be solved by techniques such as Reverse Routing Header [I-D.thubert-nemo-reverse-routing-header] (Thubert, P. and M. Molteni, “IPv6 Reverse Routing Header and its application to Mobile Networks,” February 2007.).

Battery (B): The Battery (B) flag is indicates that a parent in the tree operates on batteries, an indication of a costly operation. It is set by a mobile router which operates on battery and when set, it is left set as it is propagated down the tree.

Reserved: 5-bit unsigned integer set to 0 by the clusterhead.

Sequence Number: 8-bit unsigned integer set by the clusterhead and incremented with each new TIO it sends on a link and propagated with no change down the tree.

TreePreference: 8-bit unsigned integer set by the clusterhead to its preference and unchanged at propagation. Default is 0 (lowest preference). The tree preference provides a mechanism to engineer the mesh of mobile routers, for instance indicating the most preferred home gateway or the communication ship in a fleet at sea.

BootTimeRandom: A random value computed at boot time and recomputed in case of a duplication with another Attachment Router. 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 Mobile Router at propagation time.

Preference: The administrative preference of that (mobile) Access Router. Default is 0. 255 is the highest possible preference. Set by each Mobile Router 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 at least 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.

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 Mobile Router. This is the result of a CRC-32c computation on a bit string obtained by appending the received value and the Mobile Router 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. This value is set by the clusterhead. The global IPv6 home address of the clusterhead can be used.

The following values MUST not change during the propagation of the TIO down the tree: Type, Length, G, H, TreePreference, TreeDelay and TreeID. All other fields of the TIO are updated at each hop of the propagation.

4.2. TIO suboptions

TOC

In addition to the minimum options presented in the base option, a number of suboptions are defined for the TIO:

4.2.1. Format

TOC

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Subopt. Type | Subopt Length | Suboption Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: TIO suboption generic format

Suboption Type: 8-bit identifier of the type of mobility option. When processing a TIO containing a suboption for which the suboption Type value is not recognized by the receiver, the receiver MUST silently ignore and skip over the suboption, correctly handling any remaining options in the message.

Suboption Length: 8-bit unsigned integer, representing the length in octets of the suboption, not including the suboption Type and Length fields.

Suboption Data: A variable length field that contains data specific to the option.

The following subsections specify the TIO suboptions which are currently defined for use in the Mobility Header.
Implementations MUST silently ignore any TIO suboptions options that they do not understand.
TIO suboptions may have alignment requirements. Following the convention in IPv6, these options are aligned in a packet so that multi-octet values within the Option Data field of each option fall on natural boundaries (i.e., fields of width n octets are placed at an integer multiple of n octets from the start of the header, for n = 1, 2, 4, or 8).

4.2.2. Pad1

TOC

The Pad1 suboption does not have any alignment requirements. Its format is as follows:

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|   Type = 0    |
+-+-+-+-+-+-+-+-+
```

Figure 3: Pad 1

NOTE! the format of the Pad1 option is a special case - it has neither Option Length nor Option Data fields.
The Pad1 option is used to insert one octet of padding in the TIO to enable suboptions alignment. If more than one octet of padding is required, the PadN option, described next, should be used rather than multiple Pad1 options.

4.2.3. PadN

TOC

The PadN option does not have any alignment requirements. Its format is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
|   Type = 1    | Subopt Length | Subopt Data
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
```

Figure 4: Pad N

The PadN option is used to insert two or more octets of padding in the TIO to enable suboptions alignment. For N (N > 1) octets of padding, the Option Length field contains the value N-2, and the Option Data consists of N-2 zero-valued octets. PadN Option data MUST be ignored by the receiver.

TOC

4.2.4. Bandwidth Suboption

This suboption carries the maximum bandwidth available up the tree via a specific parent. It is the lowest speed of the links on the way and does not reflect the actual use of those links in run time. The value is expressed in the log base 2 of the speed, expressed in bps. The Bandwidth suboption does not have any alignment requirements. Its format is as follows:

```
 0                   1                   2
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---------------+
|   Type = 2    | Length = 1    | Bandwidth     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---------------+
```

Figure 5: Bandwidth Suboption

Type: Set to 2 for the Bandwidth suboption.

Length: Set to 1 for the Bandwidth suboption.

Bandwidth: 8-bit unsigned integer. The Log2 of the speed of the path expressed in bps. The clusterhead initializes that field using the speed of the link to the Access Router to which it is attached or 0xFF if it is floating. An attached MR propagates it as the minimum of the Bandwidth as received in the TIO from the parent and the access speed between the MR and the parent. As a result, the value received from a candidate AR is that of the bottleneck between that AR and the wire access.

4.2.5. Stable time Suboption

TOC

This suboption carries an indicator of the stability of a network. This indicator is the time since the branch to which the MR is attached has remained unchanged. The value is expressed in the log base 2 of that duration, expressed in milliseconds. The Stable time suboption does not have any alignment requirements. Its format is as follows:

```
 0                   1                   2
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---------------+
|   Type = 3    | Length = 1    | Stable time   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---------------+
```

Figure 6: Stable time

Type: Set to 3 for the Stable time suboption.

Length: Set to 1 for the Stable time suboption.

Stable time: 8-bit unsigned integer. The Log2 of the time since the last change in the attachment branch, expressed in milliseconds. This is set by the MR as it propagates the TIO down the tree, indicating for how long the PathDigest in the TIO from its parent has remained unchanged.

4.2.6. Tree Group ID Suboption

TOC

This suboption carries the Group ID for the tree. It is set by the clusterhead and is left unchanged by the MR that propagates the TIO down the tree. The Tree Group ID Suboption has an alignment requirement of 8n+6. Its format is as follows:

```
 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 = 4    | Length = 16   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                           Tree                                |
+                         Group ID                              +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Tree Group ID Suboption

Type: 8-bit unsigned integer. Its value is 4 for the Tree Group ID suboption.

Length: 8-bit unsigned integer. Its value is 16 for the Tree Group ID suboption.

Tree Group ID: 128-bit unsigned integer which identify a group for a tree. This value is set by the clusterhead. It can be set administratively, for instance to an IPv6 multicast group.

4.2.7. Path Free Medium Time Suboption

TOC

This suboption carries the Free Medium Time available up the tree via a specific parent at a given point of time. It is an indication of whether bandwidth is available to place VoIP calls for instance. As defined by the Quality of Service (QoS) Task Group of the Wi-Fi Alliance, the Medium Time describes the amount of time admitted to access the medium, in units of 32 microsecond periods per second. The Free Medium Time is the amount of time left the medium, in other words ((1000000/32) - SIGMA(MT)). The Path Free Medium Time is the lowest available Free Medium Time along the way and it reflects the actual use of those links in run time. The Path Free Medium Time suboption does not have any alignment requirements. Its format is as follows:

```
 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 = 5    | Length = 2    |     Path Free Medium Time     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Path Free Medium Time Suboption

Type:
Set to 5 for the Path Free Medium Time Suboption.

Length: Set to 2 for the Path Free Medium Time Suboption.

Path Free MT: 16-bit unsigned integer. The amount of Medium Time that is available along the path to the clusterhead in units of 32 microsecond periods per second. The clusterhead initializes that field to the Free MT on the link where the TIO is issued. An attached MR propagates it as the minimum of the Path Free MT as received in the TIO from the parent and the Path Free MT on the link on which the TIO is propagated. As a result, the value received from a candidate AR is that of the bottleneck between that AR and the clusterhead.

4.2.8. Uniform Path Metric Suboption

TOC

This suboption carries the Uniform Path Metric (UPM) for the path along the tree. It is set to zero by the clusterhead and incremented as the TIO is propagated down the tree. The Uniform Path Metric Suboption has an alignment requirement of 4n+2. Its format is as follows:

```
 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 = 6    | Length = 4    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Uniform Path Metric                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Uniform Path Metric Suboption

Type: 8-bit unsigned integer. Its value is 6 for this suboption.

Length: 8-bit unsigned integer. Its value is 4 for this suboption.

Uniform Path Metric: 32-bit unsigned integer aggregating the cost of radio hops.

5. Tree Discovery

TOC

Tree Discovery is a form of distance vector protocol for use in wireless mesh networks. TD locates the nearest exit and forms a Directed Acyclic Graphs towards that exit, usually a tree. TD enables Mobile Routers to implement different policies for selecting their preferred parent in the Tree by introducing the concept of plug-in, and does not specify the plug-in operation. Rather, TD specifies a set of rules to be implemented by all plug-ins to ensure interoperation. TD also standardizes the format that is used to advertize the most common information that is used by the plug-ins in order to perform the parent selection.
One of these information, the tree depth, is used by Tree Discovery to provide loop avoidance between plug-ins even if they implement different policies, and even if these policies do not use the depth as a routing metric. For instance, a MR might use the Uniform Path Metric to select the nearest exit and the best parent from the standpoint of that metric. Once attached to that parent, the MR exposes a depth which is incremented from the parent's depth, and that depth provides a comparable basis with routers which would not use UPM at all.
In order organize and maintain loopless structure, the selection plug-ins in the Mobile Routers MUST obey to the following rules and definitions:

1. A Mobile Router that is not attached to an Attachment Router is the Nemo clusterhead of its own floating tree. It's depth is 1. A Mobile Router will end up in that situation when it looses its current parent and there is no alternate parent that it can attach to. In that case, the MR SHOULD remember the treeID and the sequence counter in the TIO of the lost parent for a period of time which covers multiple TIO.

2. A Mobile Router that is attached to an Attachment Router that does not support TIO, is the clusterhead of its own grounded tree. It's depth is 1.

3. A router sending a RA without TIO is considered a grounded Attachment Router at depth 0.

4. The Nemo clusterhead of a tree exposes the tree in the Router Advertisement Tree Information Option and Mobile Routers propagate the TIO down the tree with the RAs that they forward over their ingress links.

5. A Mobile Router that is already part of a tree MAY move at any time and with no delay in order to be as close or closer to the clusterhead of its current tree - i.e. as long as that does not augment its own tree depth. But A Mobile Router MUST NOT move

down the tree that it is attached to. Mobile Routers MUST ignore RAs that are received from other routers located at the same depth or deeper within the same tree.

6. A Mobile Router may move from its current tree into any different tree at any time and whatever the depth it reaches in the new tree, but it may have to wait for a Tree Hop timer to elapse in order to do so. A MR SHOULD NOT join a previous tree (identified by its treeID) unless the sequence number in the TIO was incremented since the MR left that tree, indicating that the candidate parent was not attached behind this MR and kept getting subsequent TIOs from the same tree. The Mobile Router MAY join that other tree if it is more preferable for reasons of connectivity, configured preference, free Medium Time, size, security, bandwidth, tree depth, or whatever metrics the Mobile Router cares to use.

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

8. When A Mobile Router joins a tree, moves within its tree, or when it receives a modified TIO from its current attachment router, the Mobile Router 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 Mobile Router 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 Mobile Router fails to reach its Home Agent, it MAY attempt to roll back the movement or to retry the Home Agent 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

TOC

The tree selection is implementation and algorithm dependent. In order to limit erratic movements, and all metrics being equal, Mobile Routers SHOULD stick to their previous selection. Also, Mobile Routers SHOULD provide a mean to filter out candidate Attachment Routers whose availability is detected as fluctuating, at least when more stable

choices are available. For instance, the Mobile Router MAY place the failed Attachment Router in a Hold Down mode that ensures that the Attachment Router will not be reused for a given period of time.
The known trees are associated with the Attachment Router that advertises them and kept in a list by extending the Default Router List. DRL entries are extended to store the information received from the last TIO. These entries are managed by states and timers 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.
A Mobile Router SHOULD verify that bidirectional connectivity is available with a candidate Attachment Router before it attaches to that candidate. Some layer 2 such as 802.11 infrastructure mode will provide for this, while others such as 802.11 adhoc mode will not. If the layer 2 does not guarantee the bidirectional connectivity, then the MR needs to make sure that it can reach the AR. This can be achieved using Neighbor Sollicitation and refraining from attaching to an AR for which no neighbor cache exists, or the state is still INCOMPLETE.

5.2. Sub-tree mobility

TOC

It might be perceived as beneficial for a sub-tree to move as a whole. The way it would work is for a Mobile Router to stay clusterhead even if itself is attached into a parent tree. But the loop avoidance is based on the knowledge of the tree that the Mobile Router visit, preventing a Mobile Router 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-Mobile Router 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-Mobile Router of a tree that attaches to a fixed Access Router. That root-Mobile Router might omit to consider a TIO that comes from the new Attachment Router and decide to stay root, in order to keep the tree consistency from the nested Mobile Routers standpoint. This does not create loops, even if the path vector is not present

---

TOC

5.3. Administrative depth

When the tree is formed under a common administration, or when a Mobile Router performs a certain role within a community, it might be beneficial to associate a range of acceptable depth with that MR. For instance, a MR that has limited battery should be a leaf unless there is no other choice, and thus expose an exagerated depth. On the other hane, a MR that is designed for backhaul should operate in a low range of depth.
With Tree Discovery, a MR has to expose a depth that is incremented from its parent's depth as receive in the TIO. In particular, a MR might expose a depth which is incremented by more than one from its parent's depth, in order to fit in its own administrative range. So a depth of N does not mean that there is precisely N Mobile Routers on the way, but at most N.

5.4. DRL entries states and stability

TOC

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

> **Current** This Attachment Router is used for roaming
>
> **Candidate** This Attachment Router can be used for roaming.
>
> **Held-Up** This Attachment Router can not be used till tree hop timer elapses. This does not occur for a fixed Attachment Router that does not send a TIO since the tree delay is null in that case.
>
> **Held-Down** This Attachment Router 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 Attachment Router can not be used till its next RA.

5.4.1. Held-Up

TOC

This state is managed by the tree Hop timer, it serves 2 purposes:

> Delay the reattachment of a sub-tree that has been forced to detach. This is not as safe as the use of the sequence but still covers that when a sub-tree has detached, the Router Advertisement - Tree

> Information Option that is initiated by the new clusterhead has spread down the sub-tree so that two different trees have formed.
>
> Limit Router Advertisement - Tree Information Option 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 Router Advertisement - Tree Information Option. The Mobile Router joins the tree by selecting the source of the RA message as its attachment router (default gateway) and propagating the TIO accordingly.
When a new tree is discovered, the candidate Attachment Router that advertises the new tree is placed in a held up state for the duration of a Tree Hop timer. If the new Attachment Router is more preferable than the current one, the Mobile Router expects to jump and becomes unstable.
A Mobile Router that is unstable may discover other Attachment Routers 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 Mobile Router 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 Attachment Router 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.

5.4.2. Held-Down

TOC

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 Attachment Router disappears (upon expiration timer), and when an Attachment Router is tried but can not reach the Home Agent (upon expiration of another Attachment Router, or upon tree hop for that Attachment Router).
An Attachment Router that is held down is not considered for the purpose of roaming. When the hold down timer elapses, the Attachment Router is removed from the DRL.

5.4.3. Collision

TOC

A race condition occurs if 2 Mobile Routers send Router Advertisement - Tree Information Option at the same time and wish to join each other.

This might happen between routers at a same depth, or routers which act as clusterhead of their own tree. In order to detect the situation, Mobile Routers time stamp the sending of Router Advertisement - Tree Information Option. Any Router Advertisement - Tree Information Option 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 clusterhead (tree) preference, the remaining 24 bits is a boot time computed random.

A Mobile Router that decides to join an Attachment Router will do so between (Attachment Router depth) and (Attachment Router depth + 1) times the Attachment Router tree delay. But since a Mobile Router is unstable as soon as it receives the Router Advertisement - Tree Information Option from the preferred Attachment Router, it will restrain from sending a Router Advertisement - Tree Information Option 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 Attachment Router and the Mobile Router, 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 Attachment Router to define the duration of risk window.

There is risk of a collision when a Mobile Router receives an RA, for an other mobile router that is more preferable than the current Attachment Router, within the risk window. In the face of a potential collision, the Mobile Router with the lowest extended preference processes the Router Advertisement - Tree Information Option 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.

5.4.4. Instability

TOC

A Mobile Router is instable when it is prepared to move shortly to another Attachment Router. This happens typically when the Mobile Router has selected a more preferred candidate Attachment Router and has to wait for the tree hop timer to elapse before roaming. Instability may also occur when the current Attachment Router is lost and the next best is still held up. Instability is resolved when the tree hop timer of all the Attachment Router (s) causing instability elapse. Such Attachment Router is changes state to Current or Held-Down.

Instability is transient (in the order of tree hop timers). When a Mobile Router is unstable, it MUST NOT send RAs with TIO. This avoids loops when Mobile Router A wishes to attach to Mobile Router B and Mobile Router B wishes to attach to Mobile Router A. Unless RA cross (see Collision section), a Mobile Router receives TIO from stable Attachment Routers, which do not plan to attach to itself, so the Mobile Router can safely attach to them.

5.4.5. Density

TOC

In a dense environment, it is useless that all routers that can provide backhauling service actually do so; in practice, limiting the number of routers that accept visitors saves memory in the visitors and reduces the cost of signalling. Also, limiting the number of nodes (mobile routers that is) in the tree improves the multicast operations. Algorithms such a Trickle could be used by a Mobile Router to decide to stop providing its access services for visitors if there are a number of neighboring routers that provide similar services. The simplest abstraction of such similarity is that of multiple routers advertising a same depth, though such a simple similarity does not address the specifics of a router selection in the plugins. In a more general fashion, a Mobile Router can associate the concept of similarity with the characteristics of its own attachment router selection plug in. The Trickle algorithm must be tuned in such a fashion that the susceptibility to stop advertising services is inversely proportional to the number of nodes attached to the Mobile Router, and that the susceptibility to restore services is also inversely proportional to the time it has been suspending those services. Once a router suspends its services, it should do so for a period of time that is exponentially growing each time the decision is made again to keep suspending thise services. That period is interrupted if a neighbor is found that does not have a parent.

5.5. Legacy Routers

TOC

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 Mobile Routers from such a legacy router.

TOC

6. Directed Acyclic Graph Discovery

Tree Discovery builds trees, which are a specific form of a Directed Acyclic Graph (DAG). In a more general Fashion, TD can be adapted to form DAGs, oriented towards the clusterhead. This is DAG Discovery. In Section 5.3 (Administrative depth), TD enables a given MR to expose a depth that is incremented by more than one with regards to its parent. When it does so, a MR can elect a number of alternate parents as feasible successors. A feasible successor belongs to the same tree as the MR parent, and has a depth that is less than that of the MR. The links MR to feasible successors complete the tree as built by TD into a DAG towards the clusterhead. The DAG enables alternate exit paths for a multihomed Mobile Router.

---

7. IANA Considerations

TOC

Section 4.1 (TIO base option). requires the definition of a new option to Neighbor discovery (Narten, T., Nordmark, E., Simpson, W., and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” September 2007.) [RFC4861] messages, the Router Advertisement - Tree Information Option (RA-TIO). The Router Advertisement - Tree Information Option has been assigned the value TBD within the numbering space for IPv6 Neighbor Discovery Option Formats.

---

8. Security Considerations

TOC

At the current level of this draft, the TIO bears the security level of the RA and the link. Nothing is added to it. A deeper threat analysis would be required to eventually propose additional security.

---

9. Contributors

TOC

The editor recognizes the contributions by:

> Caroline Bontoux from Fortinet,
>
> Nicolas Montavont from Univerity Louis Pasteur,
>
> Ben McCarthy from Lancaster University,

Patrick Wetterwald and JP Vasseur from Cisco.

10. Acknowledgments

TOC

The editor wishes to thank Marco Molteni for his early participation to this design and the review of the document, Massimo Villari for his early work on simulation and research on the subject and Julien Abeille for his advanced participation in simulation and real testing. Also the authors wish to thank Christopher Dearlove for his suggestion for additional protection against loops in TD, and Philip Levis, David Cueller and Jonathan Hui for the suggestions about Trickle. This work is also based on prior publications, in particular HMRA (Cho, H., "Hierarchical Mobile Router Advertisement for nested mobile networks," January 2004.) [I-D.cho-nemo-hmra] by Hosik Cho and Eun-Kyoung Paik from Seoul National University and other non IETF publications coauthored with Thierry Ernst and Thomas Noel. Finally, the editor heartily thanks Marcelo Bagnulo Braun and Teco Boot for their very constructive reviews.

11. References

TOC

11.1. Normative Reference

TOC

[RFC4861]	Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, September 2007 (TXT).
[RFC3775]	Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004 (TXT).
[RFC3963]	Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, January 2005 (TXT).
[I-D.ietf-nemo-terminology]	Ernst, T. and H. Lach, "Network Mobility Support Terminology," draft-ietf-nemo-terminology-06 (work in progress), November 2006 (TXT).
[RFC4191]	Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes," RFC 4191, November 2005 (TXT).

11.2. Informative Reference

TOC

[I-D.cho-nemo-hmra]	Cho, H., "Hierarchical Mobile Router Advertisement for nested mobile networks," draft-cho-nemo-hmra-00 (work in progress), January 2004 (TXT).
[I-D.ietf-nemo-multihoming-issues]	Ng, C., "Analysis of Multihoming in Network Mobility Support," draft-ietf-nemo-multihoming-issues-07 (work in progress), February 2007 (TXT).
[I-D.thubert-nemo-reverse-routing-header]	Thubert, P. and M. Molteni, "IPv6 Reverse Routing Header and its application to Mobile Networks," draft-thubert-nemo-reverse-routing-header-07 (work in progress), February 2007 (TXT).

Author's Address

TOC

	Pascal Thubert (editor)
	Cisco Systems
	Village d'Entreprises Green Side
	400, Avenue de Roumanille
	Batiment T3
	Biot - Sophia Antipolis 06410
	FRANCE
Phone:	+33 497 23 26 34
Email:	pthubert@cisco.com