

NEMO Working Group  
Internet-Draft  
Expires: October 22, 2006

P. Thubert  
Cisco  
C. Bontoux  
Fortinet  
N. Montavont  
LSIIT - ULP  
April 20, 2006

Nested Nemo Tree Discovery  
draft-thubert-tree-discovery-03.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 22, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The purpose of this paper is to describe a minimum set of features that extends the Nemo basic support [4] 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.

Internet-Draft

TD

April 2006

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Terms and Abbreviations . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Motivations . . . . .	<a href="#">5</a>
<a href="#">3.1</a>	Multi-Homed nested mobile network . . . . .	<a href="#">5</a>
<a href="#">3.2</a>	Loops in nested Nemo . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Router Advertisement extensions . . . . .	<a href="#">8</a>
<a href="#">4.1</a>	Router Advertisement message . . . . .	<a href="#">8</a>
<a href="#">4.2</a>	Tree Information Option . . . . .	<a href="#">8</a>
<a href="#">4.3</a>	TIO suboption . . . . .	<a href="#">11</a>
<a href="#">4.3.1</a>	Format . . . . .	<a href="#">11</a>
<a href="#">4.3.2</a>	Pad1 . . . . .	<a href="#">11</a>
<a href="#">4.3.3</a>	PadN . . . . .	<a href="#">12</a>
<a href="#">4.3.4</a>	Bandwidth Suboption . . . . .	<a href="#">12</a>
<a href="#">4.3.5</a>	Stable time Suboption . . . . .	<a href="#">13</a>
<a href="#">4.3.6</a>	Tree Group ID Suboption . . . . .	<a href="#">13</a>
<a href="#">5.</a>	Tree Discovery . . . . .	<a href="#">15</a>
<a href="#">5.1</a>	tree selection . . . . .	<a href="#">16</a>
<a href="#">5.2</a>	Sub-tree mobility . . . . .	<a href="#">16</a>
<a href="#">5.3</a>	Administrative depth . . . . .	<a href="#">17</a>
<a href="#">5.4</a>	DRL entries states and stability . . . . .	<a href="#">17</a>
<a href="#">5.4.1</a>	Held-Up . . . . .	<a href="#">18</a>
<a href="#">5.4.2</a>	Held-Down . . . . .	<a href="#">19</a>
<a href="#">5.4.3</a>	Collision . . . . .	<a href="#">19</a>
<a href="#">5.4.4</a>	Instability . . . . .	<a href="#">20</a>
<a href="#">5.5</a>	Legacy Routers . . . . .	<a href="#">20</a>
<a href="#">6.</a>	Directed Acyclic Graph Discovery . . . . .	<a href="#">20</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">21</a>
<a href="#">8.</a>	Security Considerations . . . . .	<a href="#">21</a>
<a href="#">9.</a>	Changes . . . . .	<a href="#">21</a>
<a href="#">9.1</a>	Changes from version 00 to 01 . . . . .	<a href="#">21</a>
<a href="#">9.2</a>	Changes from version 01 to 02 . . . . .	<a href="#">21</a>
<a href="#">9.3</a>	Changes from version 02 to 03 . . . . .	<a href="#">21</a>

<a href="#">10.</a>	Acknowledgments . . . . .	<a href="#">21</a>
<a href="#">11.</a>	References . . . . .	<a href="#">23</a>
<a href="#">11.1</a>	Normative Reference . . . . .	<a href="#">23</a>
<a href="#">11.2</a>	Informative Reference . . . . .	<a href="#">23</a>

Thubert, et al.	Expires October 22, 2006	[Page 2]
-----------------	--------------------------	----------

---

Internet-Draft	TD	April 2006
----------------	----	------------

Authors' Addresses . . . . .	<a href="#">24</a>
Intellectual Property and Copyright Statements . . . . .	<a href="#">25</a>

## 1. Introduction

As per Nemo Basic support [4], 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 [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 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

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 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.

### [3.](#) Motivations

#### [3.1](#) Multi-Homed nested mobile network

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 [\[9\]](#) 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 an 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 [\[7\]](#) 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.

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

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.



New extensions of Router Advertisement are proposed to distribute the knowledge of the Mobile Router 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 Mobile Router 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 Mobile Router or not.

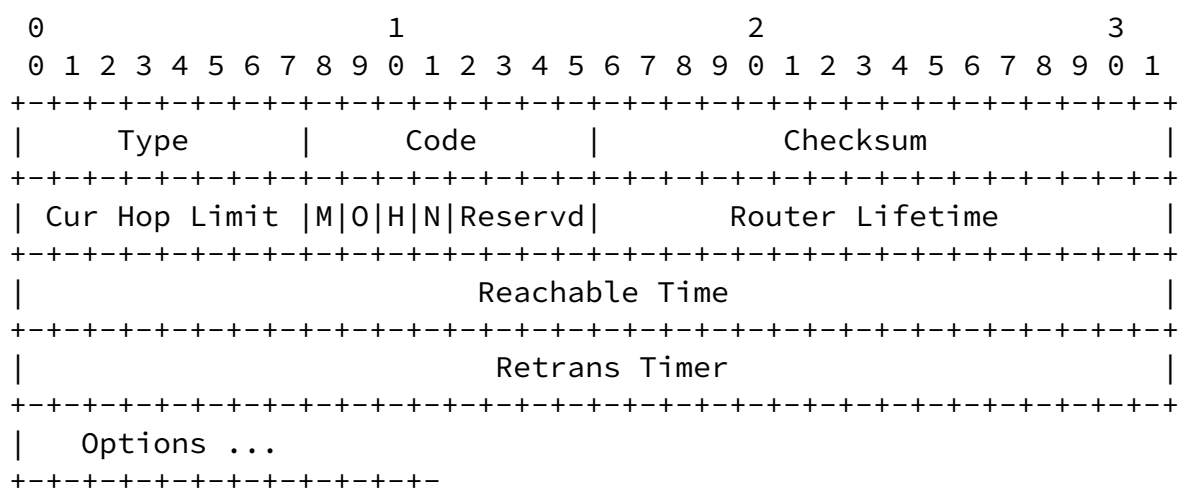


Figure 1: Router Advertisement

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 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.

The 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.

A TIO can also be used by Mobile Network Nodes to select their best default router. If the default router of a non-Mobile Router sends

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

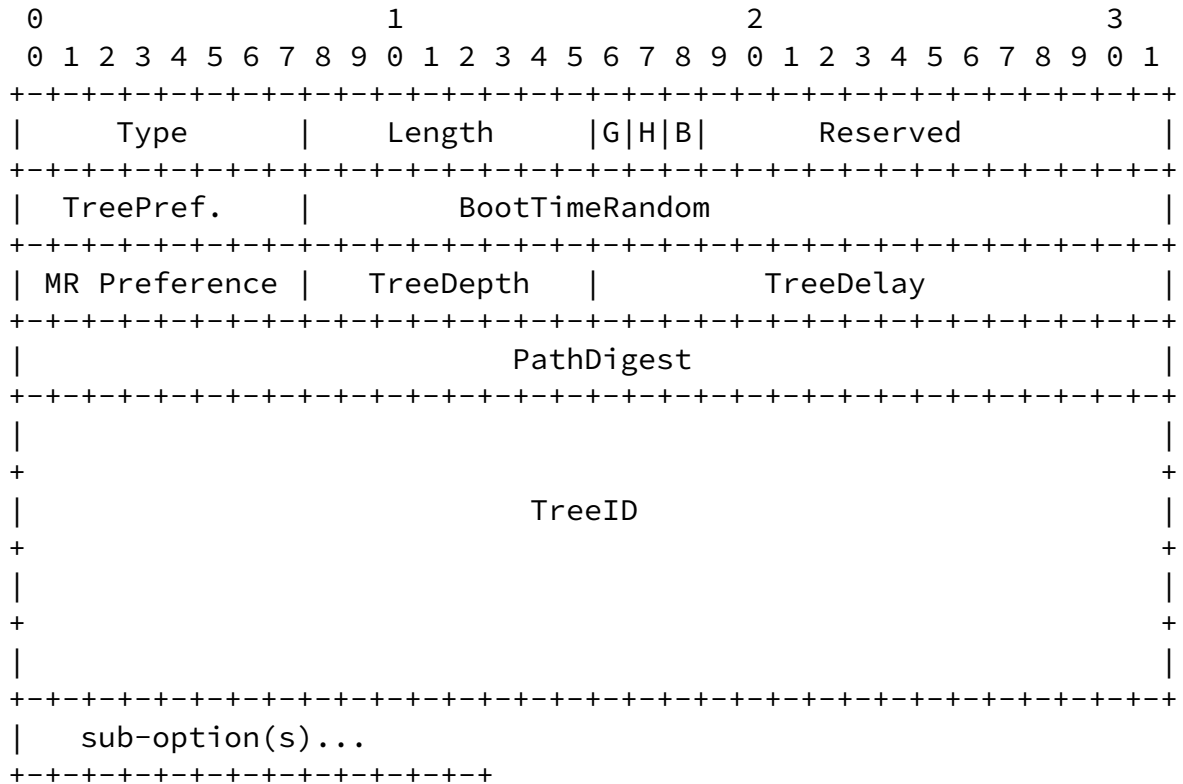


Figure 2: RA Tree Information Option

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 clusterhead is attached to its home network.

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.

Internet-Draft

TD

April 2006

Reserved: 13-bit unsigned integer set to 0 by the clusterhead.

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

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.

### [4.3](#) TIO suboption

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

#### [4.3.1](#) Format

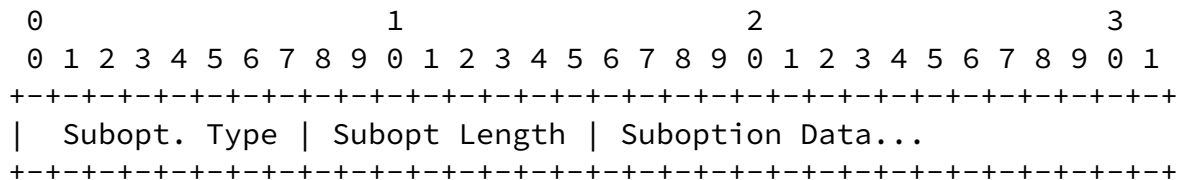


Figure 3: TIO suboption generic format

**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  $n = 1, 2, 4, \text{ or } 8$ ).

#### [4.3.2](#) Pad1

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

Thubert, et al. Expires October 22, 2006 [Page 11]

---

Internet-Draft TD April 2006

```

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

Figure 4: 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.

#### [4.3.3](#) PadN

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 5: 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.

#### 4.3.4 Bandwidth Suboption

This suboption carries the bandwidth available up the tree via a specific parent. 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 6: 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.

#### 4.3.5 Stable time Suboption



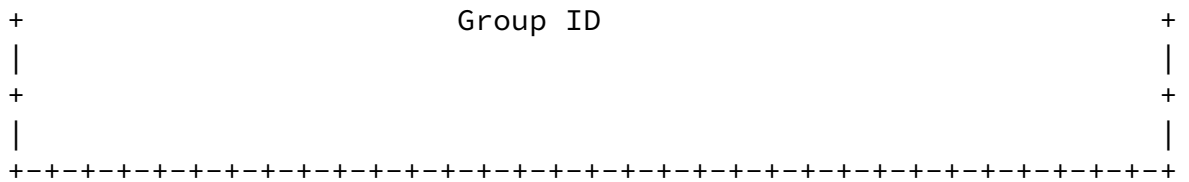


Figure 8: Tree Group ID Suboption

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.

## 5. Tree Discovery

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



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.
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 get closer to the clusterhead of its current tree - i.e. in order to reduce 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 deeper or at the same depth 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. The Mobile Router 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 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

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 Solicitation and refraining from attaching to an AR for which no neighbor cache exists, or the state is still INCOMPLETE.

### [5.2](#) Sub-tree mobility

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

### [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 exaggerated 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

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

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 allows to make sure 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  $0 \leq random < 1$ ). It is randomized in order to limit collisions and synchronizations.

#### [5.4.2](#) Held-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 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

A race condition occurs if 2 Mobile Routers send Router Advertisement - Tree Information Option at the same time and wish to join each other. 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 TI0. The first byte is the clusterhead preference, then the router own preference (default is 0 for both), the remaining 16 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

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.

## [5.5](#) 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 Mobile Routers from such a legacy router.

## [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](#), 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

[Section 4.2.](#) requires the definition of a new option to Neighbor discovery [[1](#)] 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

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.](#) Changes

### [9.1](#) Changes from version 00 to 01

Added text on sub-tree mobility from the discussion with Marcelo.

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

### [9.2](#) Changes from version 01 to 02

Improved text on instability

Changed the formula for the 4 bytes number used in collision avoidance

### [9.3](#) Changes from version 02 to 03

Added suboptions for tree group, stable time and bandwidth.

Added administrative depth and increment by more than 1.

Added words on bidirectional check using ND.

Added DAG discovery.

## [10.](#) 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 [\[8\]](#) 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.

## 11. References

### 11.1 Normative Reference

- [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", [RFC 3775](#), June 2004.
- [4] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", [RFC 3963](#), January 2005.
- [5] Ernst, T., "Network Mobility Support Goals and Requirements", [draft-ietf-nemo-requirements-05](#) (work in progress), October 2005.
- [6] Ernst, T. and H. Lach, "Network Mobility Support Terminology", [draft-ietf-nemo-terminology-05](#) (work in progress), March 2006.
- [7] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", [draft-ietf-ipv6-router-selection-07](#) (work in progress), January 2005.

### 11.2 Informative Reference

- [8] Cho, H., "Hierarchical Mobile Router Advertisement for nested mobile networks", [draft-cho-nemo-hmra-00](#) (work in progress), January 2004.
- [9] Ng, C., "Analysis of Multihoming in Network Mobility Support", [draft-ietf-nemo-multihoming-issues-05](#) (work in progress), February 2006.

Internet-Draft

TD

April 2006

## 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

Caroline Bontoux  
Fortinet  
Sophia Antipolis  
Biot 06410  
FRANCE

Email: cbontoux@fortinet.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/>

Internet-Draft

TD

April 2006

## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

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