OLSR Trees: A Simple Clustering Mechanism for OLSR

Emmanuel Baccelli

OLSR Trees: A Simple Clustering Mechanism for OLSR

Emmanuel Baccelli
Hipercom Project, INRIA Rocquencourt
78153 Le Chesnay Cedex, FRANCE
Email: Emmanuel.Baccelli@inria.fr

***Abstract*— The main ad hoc routing protocols that were proposed generally provide only flat networks. However the Internet has always been of a hierarchical nature, for scalability and manageability reasons. This paper therefore introduces a simple mechanism providing dynamic clustering with OLSR, one of the MANET routing solutions, chosen for its ease of integration in the Internet infrastructure. This clustering can have many different applications. This work describes how it can be used to provide hierarchical routing with OLSR. However, it is not limited to this use.**

I. Introduction

While the main ad hoc routing solutions OLSR [1], AODV [6], DSR [8], and TBRPF [7] generally provide only flat routing, the Internet has always been hierarchical in nature. Hierarchy was introduced as a tool to cope with scalability problems, concerning both routing and managing administratively. Indeed, having several levels of hierarchy limits the growth of the routing information needed in the biggest routers in the Internet. Hierarchy enables this growth to be only logarithmic with respect to the size of the network, instead of linear. And on the other hand, when an organization grows in size, hierarchy and clustering have obvious advantages in terms of management in general. Issues due to scalability have not been entirely resolved with the main solutions that were proposed (see [1] [6] [7] [8] [2]). However, MANET routing is in dire need to address these issues, as it suffers from what is also its advantage: native mobility disturbing the Internet architecture, and decentralized wireless access incurring a lack of bandwidth limiting its flat growth.

OLSR [1], the most popular solution easily integrated in the Internet infrastructure, is no exception to this fact. This work therefore presents a mechanism providing dynamic clustering in an OLSR network, based on a technique close to the tree clustering described in [3]. This clustering can be used for different purposes: (i) to enable hierarchical routing, or (ii) to create relatively natural regions for some administrative purpose such as address (auto)configuration, security, or any other purpose needing a dynamic partitionning of the network.

The remainder of this paper is organized as follows. The next section will briefly overview OLSR, essentially very close to the widely used routing protocol OSPF [9] [10]. The clustering mechanism will then be detailed in the context of an OLSR network. And finally an application of the clustering mechanism to hierarchical routing with OLSR will be exposed, before we conclude on the matter.

II. OLSR Protocol Overview

In this section we essentially outline OLSR, keeping in mind our goal: to design a clustering mechanism that integrates in the OLSR framework as a simple extension. For further details on OLSR, or on its performance characteristics, see [1] and [4].

As a proactive link-state routing protocol, OLSR employs the periodic exchange of control messages in order to accomplish topology discovery and maintenance. This exchange results in a topology map being present in each node in the network, from which a routing table can constructed.

Basically, OLSR employs two types of control messages: HELLO messages and TC messages. HELLO messages have local scope and are exchanged periodically between neighbor nodes only, essentially tracking the status of links between neighbors. On the other hand, TC messages have larger scope and are emitted periodically to diffuse link-state information throughout the entire network.

This operation of diffusing a message to the entire network – also called flooding – is optimized in OLSR with a mechanism called MPR-flooding (see [5] for more details on this OLSR-specific technique). This optimization reduces drastically the cost of performing a flooding operation, through having each node select a minimal set of "relay nodes" (called MPRs), responsible for relaying flooded packets. As shown in Fig. 1, from the local point of view of a node flooding a packet – *i.e.* the center node in the figure – this corresponds to only the minimal number of neighbors (the black nodes) relaying the broadcast, instead of basically all the neighbors.

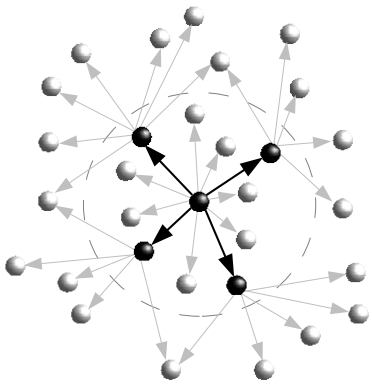

Fig. 1. Multipoint Relays of a node. A node (center) floods a message that is forwarded only by the neighbors it has selected as its MPRs (the black nodes). The range of the neighborhood of the node is depicted by the circle.

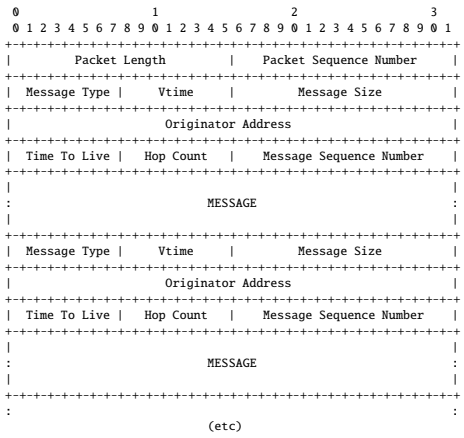

Fig. 2. Generic OLSR packet format. Each packet encapsulates several control messages into one transmission.

OLSR control traffic is transmitted in an unified packet format: this allows messages to be piggybacked together, therefore optimizing the number of transmissions overall. The OLSR packet format is shown in Fig. 2. As seen in in this figure, a packet is a collection of messages, each with individual headers. This allows the individual treatment (including flooding behavior) of each message. See [1] for further details. Note that this unified format also allows extensions to easily take advantage of the MPR flooding mechanism.

III. OLSR Tree Formation and Maintenance

The base is to pragmatically and yet optimally identify the root of trees, in other words the heads of the clusters. This must be done in a dynamic fashion, as well as the tree formation that is induced by these choices.

Taking advantage of local maximum connectivity, *i.e.* nodes that feature the most neighbors are designated cluster heads. This mechanism initially forms trees in the following way: each node selects as parent its *preferred neighbor*. A node's preferred neighbor is the neighbor which has the maximum *degree* (number of neighbors). A node which is a local maximum degree-wise (all its neighbours have lower degree) is then the *root* of its tree. Ties are broken with the classical highest ID criteria.

The network is then viewed as a *forest*, *i.e.* a collection of logical trees, as described in [3], where this mechanism is used for flooding following the branches of the trees. In this paper, we on the other hand use the clustering produced by the trees, shown in Fig. 3.

In order to enable OLSR nodes to form and maintain trees, OLSR nodes periodically exchange so-called Branch messages (in addition to usual OLSR messages). Typically a Branch message will be piggy-backed with a Hello message and have the same 1 hop scope. This approach is most scalable, since light, local and non-centralized. With a Branch message a node specifies information such as its identity (the *Node ID* field), the tree it belongs to (the *Tree ID* field) and its parent in the tree (the *Parent ID* field). The format of these messages is shown in Fig. 4. Tree options, including the description of the *Max Depth* and *Depth* fields are detailed in Section IV. The format also reserves room for eventual extensions with the *Reserved* field, unused and zeroed out, for now. Note that the IDs of the nodes are generally the IP addresses of the nodes.

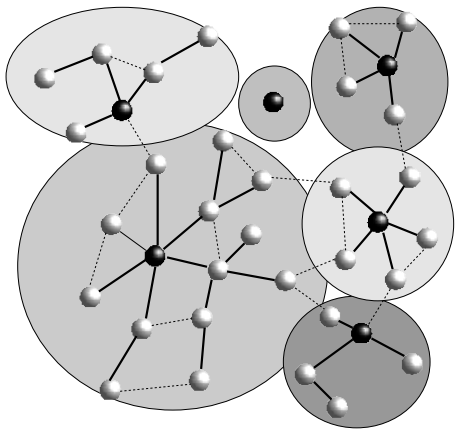

Fig. 3. Tree clustering. Roots are shown as black nodes, and branches of the trees are shown as plain links between nodes. Links that are not branches are dashed. One tree is reduced to its root, as it is disconnected from any other node.

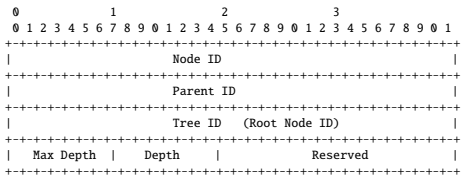

Fig. 4. OLSR Branch message format.

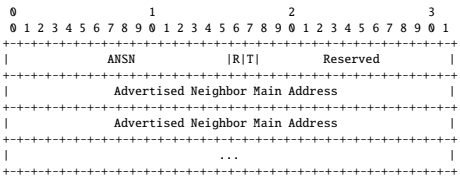

Fig. 5. OLSR TC packet format with tree options R and T.

IV. Tree Options

Several options may be provisionned in order to tune the tree mechanisms. They are discussed in the following.

A. Tree Depth Control

Roots can choose to limit the size of their tree by imposing a maximum supported depth. The idea is that a root may have to perform some extra work, as being responsible for the communication outside the tree for example. The amount of work grows with the number of nodes in the tree. A root can therefore choose to limit the extra work by imposing some limitation as to how to join its tree, based on its ressources.

This is done by the root setting the maximum depth it supports in the *Max Depth* field in the Branch messages it emits (see Fig. 4). Nodes in the tree can then be aware of this limitation and enforce it. These in turn advertize this maximum depth in their Branch message and also precise which depth they are at with the *Depth* field (the root is at depth 0). A node wanting to join the tree can then check what is the depth limitation for this tree, and therefore if it can join the tree or not.

Note that the tree depth control option can be disabled. If the root sets the *Max Depth* field to a special value (all the bits set to 1), there is no depth limitation for its tree.

B. Tree Mode Threshold

Ideally, the tree mode should appear only when the topology requires it, i.e. the MANET grows big enough. There should be a threshold above which the trees start to develop and a way to transition smoothly into the tree mode, *i.e.* a state where all the nodes in the MANET are tree-aware, sending and receiving Branch messages. This way, an application using clustering can then start being ensured that the tree structures are in place in the entire network – this may be very important to have, depending on the application. The reverse should also be made possible: below this threshold, trees should start to disappear and there should be a way to smoothly transition out of the tree mode.

This threshold can be of various nature: the size of the link state database, the frequency of TC receival or any complex equation determining if it would be beneficial to transition into or out of this hierarchical mode.

1) Transition Into Tree Mode:

When a node decides that the threshold is reached, it checks if it is in a position to be root of its tree. If it is, it starts sending Branch messages as such. A node that receives a Branch message checks if its threshold is indeed reached and if it is, it may decide to join the tree it belongs to according to the afore-mentioned rules, and start sending Branch messages too. This way, trees grow, starting from the root. Note that a root emiting Branch messages also marks the TCs it emits with setting the R bit (see Fig. 5), this signals to other nodes in the network and outside the tree, that the node is a root.

While transitioning into tree mode, some nodes may be already in tree mode while some other nodes are not.In order to signal the transition status, nodes that are in tree mode mark their TC messages as coming from the forest. This is done with root nodes setting the R bit in their TCs and other nodes setting the T bit in their TCs (see Fig. 5).

Once there are no more unmarked TCs being flooded in the MANET, the MANET is ready to shift to tree mode: all the nodes have shifted to tree mode and the tree structures are in place. Therefore the transition can happen, as smooth as possible.

If after some amount of time there are still unmarked TCs being flooded in the MANET, this either means that (i) the network is not too big after all, but rather stable at the limit of being so, or (ii) some nodes are tree-mode incapable and therefore tree mode is impossible in this MANET. In that case nodes may decide to abandon the transition into tree mode and stop sending branch messages (and marking TCs).

2) Transition Out of Tree Mode:

When a root determines that the threshold is reached, it may decide to transition back into regular mode. In that case, it will start marking its TCs with both T and R bits set. Setting both R and T bits indicate that this tree wants to revert back to not using the tree structure any more. When another root receives a TC both marked with R and T bits

set, it may check wether its threshold is reached or not and may also start to mark its own TCs with both R and T bits set.

If a state is reached where all the TCs marked with the R bit set also have the T bit set, the MANET is ready to transition back, out of tree mode, as smoothly as possible.

If after some amount of time there are still some TCs being diffused in the MANET with the R bit set but without the T bit set, this means that the network is not ready to revert. In that case roots may decide to abandon the transition out of tree mode and stop marking their TCs with T bit set.

V. Hierarchical Routing with OLSR Trees

One application of the tree structuring described above can be the introduction of hierarchical routing in OLSR, using the dynamic clustering defined by the trees. The following sections briefly describe a way to achieve that when the tree structures are in place. Note that, as mentionned in the introduction, there may be other applications that may benefit from using this clustering, and even, other ways to use OLSR trees for hierarchical routing.

A. Routing within Tree Scope

Within a tree, OLSR operates as if there was no tree, except for the following points:

1) Messages coming from a neighbor that is not in the same tree are generally not considered and not forwarded.
2) The root of a tree has the special additional role of being responsible for the communication of the tree with the rest of the MANET.
3) A node in contact with another tree must inform its own tree and especially its root.

In the following, we will describe how the restriction to tree scope is done, and how the root performs its special role. Note that routing within a tree is identical to routing with regular OLSR, and that the only difference stands in routing outside the tree.

1) Flooding within Tree Scope:

MPR selection is unaltered by the use of trees: MPRs are selected as if there were no trees. The MPR mechanism is local and therefore very scalable. What is less scalable is the diffusion by all the nodes in the network (no hierarchy) of all the link state information (i.e. TC messages).

Addressing this, the tree mode enables the flooding of TC messages by any node in a tree to be restricted to that tree. In other words: TC messages originated and flooded inside a tree remain inside this this tree i.e. they are not forwarded nor considered outside the tree: they are not forwarded beyond this tree. This is done via usual MPR flooding, with an additional rule: A node will not forward a message coming from a neighbor from another tree, except if

1) It is selected as MPR by this neighbor, AND
2) It is the first time it receives this message, AND
3) It has another neighbor that is in the same tree as this neighbor.

This rule ensures the MPR flooding will be complete inside the tree. In order to make sure that the MPR flooding completeness is not broken since MPR selection does not take into account tree segregation, border nodes just oustside the tree may relay messages between two different neighbors from the same tree (different from the border node's tree).

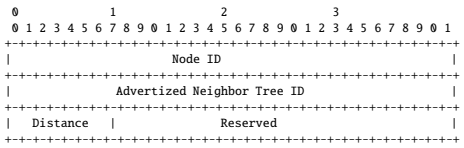

Fig. 6. OLSR Leaf packet format.

2) Leaf Nodes:

A node in contact with another tree (a node that has one or more neighbors that are not in the tree) must inform its tree and especially its root node. For each other tree this node reaches to, it can inform its tree with a so-called Leaf message specifying the roots of the other trees and its estimation of the distance between the roots (i.e. the sum of its depth in its tree and the depth of its neighbor in its own tree). The node will periodically flood this Leaf message throughout the tree, unless it has already received another Leaf message advertizing the same tree with a shorter distance estimation (and this information is still fresh enough). This way, the root and the other nodes in the tree are informed of the paths leading to any neighbor tree, and these are shortest available paths through the trees, from root to root.

Leaf messages are typically piggybacked with TC messages inside a tree and share the same scope, *i.e.* tree-scope. Their format is shown in Fig. 6. They include information such as the identity of the advertizing node (the *Node ID* field), the identity of the advertized tree (the *Advertized Neighbor Tree ID* field), or the estimated distance between the root of the tree and the root of the advertized tree (the *Distance* field).

B. Communication with Other Trees

OLSR routing and MPR flooding being restricted to a tree, something special must be done in order to distribute routing information MANET-wide, from tree to tree. This

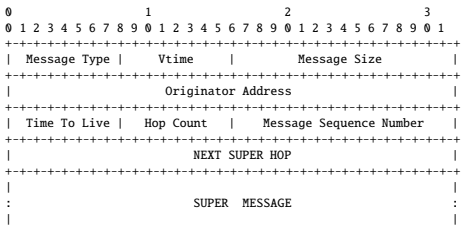

Fig. 7. OLSR Super packet format.

is the additional task of the root of a tree. In order to address this task, the root basically operates OLSR at a higher level: over the super-topology formed by the roots of trees throughout the MANET. At this level, each tree, embodied by its root, behaves as if it were a single OLSR node: a super node. Similarly to regular OLSR, these super nodes (i.e. the roots) periodically send Super-Hellos, and Super-TCs. These super-messages are the only messages to be forwarded outside a tree. This is described in the following.

1) Super Messages:

Super messages are identical to regular messages except that they feature an additional IP address in their header that indicates the next super-hop (the next root to reach). The essential difference with regular OLSR messages stands in the fact that super-messages are routed and use OLSR-established paths inside each tree, instead of being simply flooded. With hierarchical routing in place, these messages are the only messages that are forwarded outside tree scope, therefore featuring MANET scope. The format is shown in Fig. 7. All the fields are as specified in [1], except that the *Message Type* field is set to a special value indicating a super message, and the fact that the header of the message (actually the beginning of the super-message) is completed with an additional IP address specifying the next super-hop.

2) Super Hello Messages:

The root periodically sends a Super-Hello message to all the other roots it knows of via Leaf messages. Super-Hellos are unicasted and use the shortest root-to-toot paths advertized by the Leaf messages and OLSR routing/forwarding inside each tree. This way, as in OLSR, roots are informed of their super-neighborhood and can perform super-MPR selection. Super Hellos only have one super-hop scope (they are not forwarded further than the neighbor roots).

Super-Hellos are similar in functionality and format to regular Hellos messages, except they also feature the next super-hop in their header (as mentionned above). Nodes use this IP address to route the message from root to root.

3) Super TC Messages:

In addition to Super-Hellos, the root periodically sends a Super-TC message that is super-flooded (concurrent unicasts using Super-MPR and the shortest root-to-root OLSR paths) to all the roots in the network. Note that Super-TC messages therefore have a scope that is bigger than one super-hop since they are forwarded way beyond neighbor roots: throughout the whole MANET. This way, roots are informed of the whole super-topology formed by the roots.

Super-TC messages are similar in functionality and format to regular TC messages, except they also feature the next super-hop in their header (as mentionned above). Subsequent roots update this field in order to achieve super-MPR flooding over the super-topology. The format is specified in the last section.

4) Super HNA Messages:

Super-HNA messages are also periodically super-flooded by each root to all the other roots in the MANET. With the generation of a Super-HNA message, a root summarizes the link state information its cluster encompasses. This way, roots are aware of the link state information of the other trees.

Super-HNA messages are similar in functionality and format to regular HNA messages, except they also feature the next super-hop in their header (as mentionned above). They are generally piggy-backed with the generated Super-TCs. Note that it can actually be envisionned to collapse Super-TCs and Super-HNAs in only one message type that would accompish both functionalities. It was not presented here for purposes of simplicity in explaining OLSR over the super-topology.

5) Routing Beyond Tree Scope:

Being in possession of MANET-wide information with Super-HNA and Super-TC messages, a root node will then be able to route beyond tree scope. It will therefore advertise the default route inside its tree and traffic with outside the tree will transit via the root.

VI. Conclusion and Future Work

Addressing the lack of alternatives to flat networking in the main MANET routing solutions, this paper presents a dynamic clustering mechanism for OLSR [1], one of these solutions, chosen for its particular ease of integration within the Internet infrastructure. This is indeed the goal with the introduction of hierarchy in ad-hoc networking: facilitate the integration of MANETs in the Internet architecture, and address scalability issues within MANETs – issues that are

left to be completely resolved with the main solutions that are proposed (i.e. OLSR [1], AODV [6], DSR [8], and TBRPF [7]). The clustering can be used for different purposes such as routing, or administrative purposes that could benefit from the dynamic partitionning of the network into relatively natural regions. These purposes include, but are not limited to, address autoconfiguration and security. In this paper, an application of the clustering mechanism is described in order to introduce hierarchical routing with OLSR. Future work will tackle using the clustering mechanism for other applications in large MANETs such as: address autoconfiguration mechanisms, distributed security authorities and group management, and other ways to use hierarchical routing, including mechanisms using clustering to provide more stability in face of mobility.

References

[1] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, "Optimized Link State Routing Protocol," RFC 3626, http://ietf.org/rfc/rfc3626.txt, 2003.

[2] S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, http://ietf.org/rfc/rfc2501.txt, 1999.

[3] Navod Nikaein, Houda Labiod, Christian Bonnet, "DDR - Distributed Dynamic Routing Algorithm for Mobile Ad hoc Networks," MobiHOC Proceedings, 2000.

[4] T. Clausen, P. Jacquet, L. Viennot, "Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks," Proceedings of IFIP Med-Hoc-Net 2002, September 2002.

[5] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," INRIA Research Report RR-3898, March 2000.

[6] C. E. Perkins, E. M. Royer, S. R. Das, RFC 3561: "Ad Hoc On-Demand Distance Vector Routing." Internet Engineering Task Force, Request For Comments (experimental), July 2003.

[7] R. Ogier, F. Templin, M. Lewis, RFC 3684: "Topology Dissemination Based on Reverse-Path Forwarding." Internet Engineering Task Force, Request For Comments (experimental), February 2004.

[8] D. Johnson, D. Maltz, Y. Hu, draft-ietf-manet-dsr-10.txt: "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks." Internet Engineering Task Force, Internet Draft (Work in Progress), July 2004.

[9] J. Moy, "OSPF version 2," Internet Engineering Task Force RFC 2328, http://ietf.org/rfc/rfc2328.txt, 1998.

[10] C. Adjih, E. Baccelli, P. Jacquet, "Link state routing in wireless ad-hoc networks", MILCOM 2003 - IEEE Military Communications Conference, vol. 22, no. 1, pp. 1274-1279, Oct. 2003.