OLSR TREES: A SIMPLE CLUSTERING MECHANISM FOR OLSR

Emmanuel Baccelli
Hipercom Project, INRIA Rocquencourt
78153 Le Chesnay Cedex, FRANCE
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.

Keywords: Mobile Ad Hoc Networks, Clustering

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 infras-
tructure, 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.

1. 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 broad-
cast, instead of basically all the neighbors.

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

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 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.

2. 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. The Depth field indicates the distance of the node to the root. 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.

3. 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 mentioned in the introduction, there
Figure 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.

Figure 4. OLSR Branch message format.

may be other applications that may benefit from using this clustering, and even, other ways to use OLSR trees for hierarchical routing.

3.1 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.

**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 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 outside the tree may relay messages between two different neighbors from the same tree (different from the border node’s tree).

**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 advertising 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. 5. 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).

3.2 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 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.

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. 6. 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)
completed with an additional IP address specifying the next super-hop.

**Super Hello Messages.** The root periodically sends a Super-Hello message to all the other roots it knows of via Leaf messages. Super-Hello messages are unicasted and use the shortest root-to-root paths advertised 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 mentioned above). Nodes use this IP address to route the message from root to root.

**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 mentioned 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.
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 mentioned above). They are generally piggy-backed with the generated Super-TCs. Note that it can actually be envisioned to collapse Super-TCs and Super-HNAs in only one message type that would accomplish both functionalities. It was not presented here for purposes of simplicity in explaining OLSR over the super-topology.

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.

4. 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 partitioning 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


