



# OLSR SCALING WITH HIERARCHICAL ROUTING AND DYNAMIC TREE CLUSTERING

Emmanuel Baccelli

► **To cite this version:**

Emmanuel Baccelli. OLSR SCALING WITH HIERARCHICAL ROUTING AND DYNAMIC TREE CLUSTERING. IASTED International Conference on Networks and Communication Systems, Mar 2006, Chiang Mai, Thailand. 2006. <hal-00654552>

**HAL Id: hal-00654552**

**<https://hal.inria.fr/hal-00654552>**

Submitted on 22 Dec 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OLSR SCALING WITH HIERARCHICAL ROUTING AND DYNAMIC TREE CLUSTERING

Emmanuel Baccelli

Project Hipercom, INRIA Rocquencourt 78153 Le Chesnay Cedex, FRANCE

Email: Emmanuel.Baccelli@inria.fr

## ABSTRACT

The Internet uses hierarchical networking, for scalability and manageability reasons. However the main ad hoc routing solutions (OLSR [1], AODV [6], DSR [8], TBRPF [7]) only provide flat networking, and generally suffer important scalability issues. This paper therefore introduces a simple mechanism providing dynamic clustering with OLSR, one of the MANET routing solutions, chosen for its easy integration in the Internet infrastructure. The paper then describes how this dynamic clustering can be used to provide MANET hierarchical networking with OLSR.

## KEY WORDS

ad-hoc, networks, clustering, hierarchical, routing, protocol

## Introduction

The Internet's hierarchical design controls the growth of the routing information needed in the biggest routers in the Internet, enabling this growth to be only logarithmic with respect to the size of the network, instead of linear. This property enables this design to scale to the large Internet topologies we know today.

On the other hand, the main ad hoc routing solutions (such as OLSR [1], AODV [6], DSR [8], or TBRPF [7]) provide only flat routing, and these these solutions have not entirely resolved ad hoc networking's scalability issues. Indeed, MANETs (Mobile Ad hoc NETWORKS [2]) are currently quite limited in the number of nodes they can manage, due to the saturation of the wireless links by the routing protocol's control traffic when the network grows too large. However, one can anticipate that in the near future, bigger ad hoc topologies will have to be managed, as this network technology may become quite popular (we can for example imagine a metropolitan area with users carrying smart phones featuring WiFi interfaces and ad hoc routing software, which may happen quite soon). In order to be able to scale to larger MANET topologies, ad hoc routing is in dire need to use specific scalability tools such as, for instance, hierarchical networking.

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], that is then used to enable hierarchical routing in MANETs. OLSR was chosen as it is the ad hoc routing protocol currently standardized in the IETF (Internet Engineering Task Force, the body behind Internet standards) that is the most compatible with usual Internet routing : our focus here is put on MANETs that can be easily integrated in the Internet infrastructure. The remainder of the 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, its application 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 be constructed.

Basically, OLSR employs two types of control messages: HELLO messages and TC (Topology Control) 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 (MPR stands for Multi-Point Relay, 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

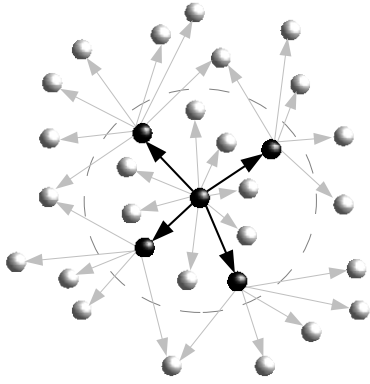


Figure 1. Multipoint Relays of a node. A node (center) floods a message that is forwarded only by the neighbors it has selected as its MPRs (the black nodes). The range of the neighborhood of the node is depicted by the circle.

minimal number of neighbors (the black nodes) relaying the broadcast, instead of basically all the neighbors.

Additionally to the base described above (TC and HELLO message exchanges and MPR mechanisms), OLSR employs two other types of messages: (i) MID messages (Multiple Interface Declaration), with which a node with multiple interfaces can declare its interfaces configuration to the other nodes in the network, and (ii) HNA messages (Hosts and Network Association), with which a node can advertise routes to networks or hosts that are outside the OLSR network. Several such different messages may be piggybacked in a single packet, in order to reduce the number of overall transmissions.

## 2 OLSR Tree Formation and Maintenance

Similarly to all the other main ad hoc routing solutions, OLSR cannot scale to large topologies. The control traffic it generates grows too big when the number of nodes in the network increases, and wireless links are quickly saturated with control traffic on its own. In order to enable OLSR to manage bigger ad hoc topologies, one (or more) specific mechanisms must be used in addition to the basic protocol. In this section, we describe such a mechanism, that makes use of dynamic clustering based on trees and hierarchical networking over this cluster architecture. In the following, each cluster will be referred to as a *tree* and each cluster head will be referred to as the *root* of its tree. A mechanism is needed to pragmatically and yet optimally designate roots. This must be done in a dynamic fashion, as well as the tree formation that is induced by these choices.

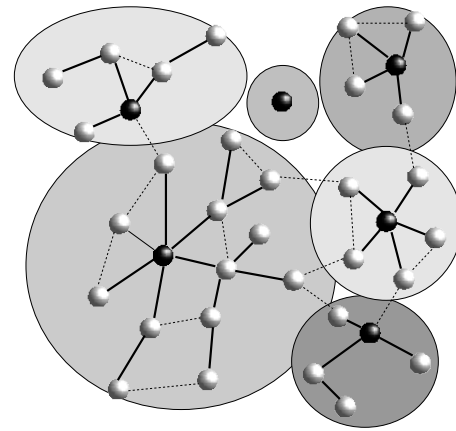


Figure 2. Tree clustering. Roots are shown as black nodes, and branches of the trees are shown as plain links between nodes. Links that are not branches are dashed. One tree is reduced to its root, as it is disconnected from any other node.

Taking advantage of local maximum connectivity, *i.e.* nodes that feature the most neighbors are designated roots. The proposed 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. 2.

In order to enable OLSR nodes to form and maintain trees, OLSR nodes periodically exchange so-called Branch messages (in addition to usual OLSR control messages). Typically a Branch message will be piggy-backed with a Hello message and have the same 1 hop scope. This approach is scalable, since it only requires neighbors to exchange messages of size and frequency that are independent of the size of the network. 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. 3. 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.

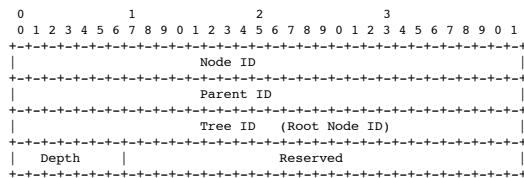


Figure 3. OLSR Branch message format.

### 3 Tree Options

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

#### 3.1 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 resources. 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. 3). Nodes in the tree can then be aware of this limitation and enforce it. These in turn advertise 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. If it cannot join the tree due to depth limitation, the node will consider joining the “next best” available tree, via the neighbour which features the “next best” (*degree, ID*) criteria (and which is not over depth limit in its tree). If none of the available trees are joinable, the node will then consider itself as root, in its own tree. 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.

#### 3.2 Tree Mode Threshold

In some cases, if the network size varies dynamically, potentially from small sizes (where flat networking is still efficient) to bigger sizes (where flat networking is not efficient anymore), it may be desirable to have a mechanism that can automatically switch between (i) usual flat networking *i.e.* the flat mode, and (ii) hierarchical networking with trees *i.e.* the tree mode. In this section we describe such a scheme. Indeed, 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 hot-switch into the tree

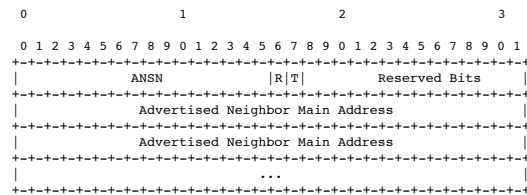


Figure 4. OLSR TC packet format, with R and T tree option bits.

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

#### 3.2.1 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 emitting Branch messages also marks the TCs it emits with setting a specific option bit, the R bit (see Fig. 4), which 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. 4). 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 hot-switching can now happen. 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).

### 3.2.2 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 whether 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. 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.

## 4 Hierarchical Routing with OLSR Trees

The tree structure described above can be used to introduce hierarchical routing in OLSR, using the dynamic clustering defined by the trees. The following sections describe a way to achieve that when the tree structures are in place.

### 4.1 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.

#### 4.1.1 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

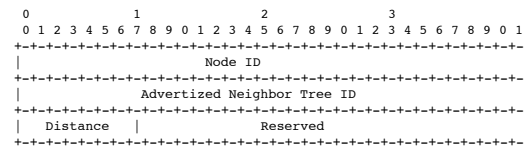


Figure 5. OLSR Leaf packet format.

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).

#### 4.1.2 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. 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).

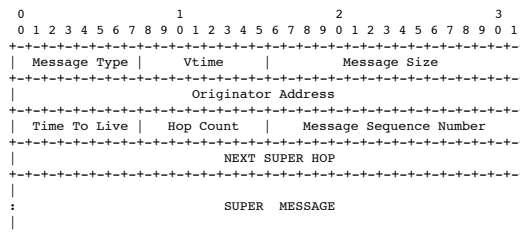


Figure 6. OLSR Super packet format.

## 4.2 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.

### 4.2.1 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. 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) is completed with an additional IP address specifying the next super-hop.

### 4.2.2 Super Hello, Super TC, and Super HNA 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-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.

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 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.

### 4.2.3 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.

## 5 Conclusions

Mobile ad hoc networking cannot scale to large topologies with the main solutions that are currently proposed (i.e. OLSR [1], AODV [6], DSR [8], and TBRPF [7]). One of the important difference between these protocols and those used in the Internet to scale to large topologies is the use of hierarchical networking, a classical and powerful scalability tool. While the Internet makes use of such a tool, MANET protocols only provide flat networking.

Thus, in order to facilitate the integration of MANETs in the Internet architecture, and to address scalability issues within MANETs, this paper presents a dynamic clustering mechanism for OLSR [1], that is applied to provide MANET hierarchical routing.

Clustering is also interesting in that it can limit the effect of micro-mobility (mobility inside a cluster) on the overall topology signalling. We note as well that this clustering mechanism could in fact also be used for different purposes, including address autoconfiguration, security, group management or any administrative purpose that could benefit from the dynamic partitioning of the network into relatively natural regions.

## References

- [1] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, "Optimized Link State Routing (OLSR) 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 (AODV) 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 (TBRPF)." 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 (DSR) Protocol for Mobile Ad Hoc Networks." Internet Engineering Task Force, Internet Draft (Work in Progress), July 2004.
- [9] J. Moy, "OSPF version 2," 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.