# MORHE: A TRANSPARENT MULTI-LEVEL ROUTING SCHEME FOR AD HOC NETWORKS

Michael Voorhaen
*PATS Research Group, University of Antwerp*
*Dept. Mathematics and Computer Science*
*Middelheimlaan 1, B-2020 Antwerpen, Belgium*
michael.voorhaen@ua.ac.be

Erwin Van de Velde
*PATS Research Group, University of Antwerp*
*Dept. Mathematics and Computer Science*
*Middelheimlaan 1, B-2020 Antwerpen, Belgium*
erwin.vandevelde@ua.ac.be

Chris Blondia
*PATS Research Group, University of Antwerp*
*Dept. Mathematics and Computer Science*
*Middelheimlaan 1, B-2020 Antwerpen, Belgium*
chris.blondia@ua.ac.be

**Abstract** This paper presents a transparent multi-level routing scheme, named MORHE, that improves the scalability of the OLSR protocol by exploiting the heterogeneous nature of nodes in the network. In our work we try to take an approach that focuses on scenarios where ad hoc technology can be applied, but where we also find nodes in the network with varying capacity. The MORHE protocol makes use of nodes which have a large capacity (e.g. more energy, larger transmission range) to build something that could best be described as an ad hoc infrastructure. Nodes are grouped in clusters that need to be interconnected by specific nodes. This implies that a node no longer needs to know the entire network topology as is the case of the OLSR protocol, but only needs to maintain routes to the nodes inside its own cluster and to the other clusters. Using this approach the signalling overhead - which is one of the main reasons why OLSR is not scalable - is greatly reduced. We also introduce a simple mobility management scheme to allow nodes to roam the different ad hoc clusters.

**Keywords:** Routing, Scalability, OLSR, Hierarchical, Click, IP Address Compacting

## Introduction

Mobile Ad-hoc NETworks (MANETs) can be used for a wide variety of applications, from small scale day-by-day extensions of infrastructured networks that allow users to communicate and have access to multimedia information, up to large scale disasters where fixed infrastructure has failed and an ad hoc network can be used as a temporary, much needed, replacement. The latter case is especially interesting for public safety users who need a system that can be deployed quickly without end user intervention and that allows access to all types of information.

Much of the current research on ad hoc has focused on flat network topologies where the nodes are assumed to have more or less the same capacities and similar needs to have access to the network. Ad hoc routing protocols like OLSR, AODV and DSR ([6], [7], [8]) are meant to allow best effort multihop communication in wireless networks while being resilient to mobility. However research has shown that flat routing protocols do not scale to large networks, since they need to maintain a seperate route for every host in the network.

In this paper we present a transparent multi-level routing scheme called MORHE (*M*ulti-level *O*LSR *R*outing using the *H*NA *E*xtension) that allows any OLSR compliant node to participate in the network without modifications. Our approach takes advantage of the heterogeneousity of the nodes that are present in the network. E.g. units that are less power constrained can be equipped with multiple interfaces, have a larger transmission range, more available bandwith or a combination of the previous. The MORHE protocol will use these nodes to form what can be best described as an ad hoc infrastructure. The ad hoc network is divided into several seperate clusters that are interconnected using the less constrained units, present in the network. During the remainder of this paper these nodes are addressed as *backbone enabled nodes* or simply backbone nodes. Using concepts that were previously defined in the OLSR standard [6] such as the HNA extension and the concept of hierarchical addressing we were able to implement a transparant scheme that allows an overlay network to be built and that minimizes overhead in the network, while still allowing nodes to roam between the different clusters. This transparancy is important since the protocol only affects the backbone enabled nodes but it does not require any fundamental changes to the OLSR protocol. Every OLSR compliant node can thus join a MORHE network.

We assume that the backbone enabled nodes are chosen before network deployment. In a public services scenario this is feasible since there the backbone enabled nodes could be police cars or fire trucks and the public service units normally only move in the vicinity of these vehicles. However the resilience to mobility that is inherent to ad hoc protocols is needed since the public service units should not be limited to staying near these vehicles for communication.

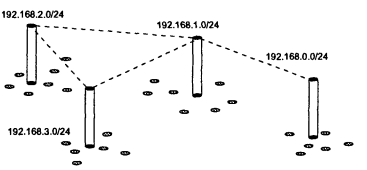


*Figure 1.* Example topology of a MORHE network

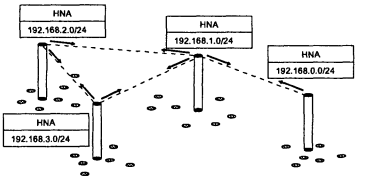


*Figure 2.* Building the overlay network using HNA messages. step 1.

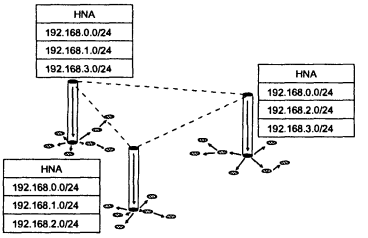


*Figure 3.* Building the overlay network using HNA messages. step 2.

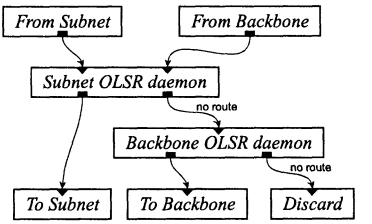


*Figure 4.* Operation of the backbone nodes

The *Hierarchical OLSR* protocol [11] improves the scalability of the OLSR protocol by adding HTC messages that are used to build a hierarchy in the network. MORHE behaves similarly to HOLSR, however it builds a hierarchical network using the HNA extension allowing it to be backwards compatible with existing OLSR implementations.

## 1. A Transparent Multi-level Routing Scheme

Figure 1 shows an example of the topology of a MORHE network with 2 levels. All the nodes in the network are grouped into smaller ad hoc clusters and each cluster is provided with a node that is backbone enabled, i.e. it has two interfaces: one for communicating in the cluster and another for communicating on the backbone. This scenario could be extended by having several levels in the backbone, or allowing all the nodes in the clusters to act as relays for their Personal Area Network (PAN) equipment. As explained in the following sections, the MORHE protocol allows for any number of levels in the network given that the address space is large enough.

## Building the overlay network

In this section we describe how the main components of the MORHE protocol work.

The backbone enabled nodes run an OLSR deamon on each of the interfaces. These two deamons communicate in a very simple setup displayed in Fig. 4: if a packet arrives at a backbone enabled node it will first attempt to look for a route to the destination on the subnet. If this failed the packet is passed to the backbone where another route lookup is performed. If a route is found the packet is forwarded, else the packet is discarded and the source node is notified that there is no route to the host.

The routes to the different clusters are advertised using Host and Network Association (HNA) packets that are described in the OLSR RFC [6]. Each backbone node periodically sends out a HNA message informing the other backbone nodes that it can reach all the nodes in the subnet that it is connected to. This is shown in Fig. 2.

The same process is repeated for the clusters: when a backbone node receives a HNA message it updates its association database. This association database is then advertised in the cluster, again using a HNA message, informing all the nodes in the cluster about what other clusters can be reached. This procedure is shown in Fig. 3.

## Mobility Management

We now propose a transparant mobility management scheme that only affects the modules that generate the HNA messages at the backbone nodes. This minimizes the impact on the nodes that are deployed in the network, i.e. each OLSR compliant node can operate without problems in this network.

The scheme is based on the proactive nature of OLSR, being that each of the MPR (Multi Point Relay) nodes advertises its neighbors at a regular basis. This way each node in the network learns of the routes to the different destinations in the network. This means that all the backbone enabled nodes also learn of the visitors that are in their subnet by looking at the IP address prefix. Each time a HNA message is generated by a backbone node it will not only advertise reachability of its own subnet, but also reachability of the visiting nodes, which it learns of by scanning its routing tables. A node that receives one of these HNA messages will then add a host specific entry to its routing table as specified by the OLSR standard.

## Minimizing the size of HNA messages

The use of HNA messages reduces the network overhead already significantly. However it is possible to use the subnet notation and the fact that all

traffic has to pass through the backbone for even further reduction of the overhead. We will propose a solution in two parts: Redundancy Check and IP Address Compacting.

**Redundancy Check.** If a backbone node has an own subnet and e.g. it has 192.168.1.0/24, it will communicate this to all other backbone nodes using HNA messages. If e.g. node 192.168.1.5 is visiting another subnet, the backbone node of that subnet will communicate the reachability of this node to all other backbone nodes. If this is the case, all other backbone nodes receiving both the HNA messages would tell their subnets that they can reach 192.168.1.0/24 and 192.168.1.5/32. It is easy to see that the information about 192.168.1.5 is redundant as 192.168.1.0/24 contains this address allready. The redundancy check finds this redundant information and deletes it, before sending information into the subnet. Of course, this information is not redundant in the routing tables of the backbone node, where it is possible that there is another gateway for 192.168.1.0/24 than for 192.168.1.5/32.
This is an easy algorithm when the information of the HNA messages of other backbone nodes is kept sorted in a table:

1 If a new element has to be inserted, search through the list to find its place (sort).

2 When the location for the element has been found, check with the previous element if that element contains the new element. If so, do not insert the new element, otherwise insert it.

3 When a element has been inserted, check if the new element contains the next element. If so, delete the next element and repeat this step. Otherwise, stop.

The second step is correct when the entries have the form ⟨ Network IP address, subnet mask ⟩ where the network IP address is the network prefix followed by all zeroes (in binary notation). All IP addresses in this subnet will have a host postfix that is not equal to zero and will come after the subnet element in the sorted list.

**IP Address Compacting.** This algorithm is applicable in any backbone node whose subnet has visiting mobile nodes. When two visiting nodes have IP addresses that differ only in the last bit, e.g.: 192.168.1.2/32 and 192.168.1.3/32, these addresses can be brought together using the subnet mask as 192.168.1.2/31. In this section, we will explain the algorithm. The same compression can be executed on the redundancy check table after the successful execution of the redundancy check and before broadcasting this information into the cluster. We will explain the case of the visitor table first.

Every time the backbone node has to update its routing and visitor tables, OLSR drops the entire routing table and all visitor information. So the algorithm must be able to reconstruct the entire visitor table, compressed, with a low cost. The algorithm works iterative and will keep a sorted visitor table.

1 Insert the first visiting node in the visitor table with subnet mask 255.255.255.255 (/32)

2 Insert another visiting node in the visitor table with subnet mask 255.255.255.255 (/32), so that the IP addresses with their subnet masks in the visitor table are ordered from the lowest to the highest address.

3 If the previous or next pair of ⟨ IP address, subnet mask ⟩ has the same subnet mask and they differ only in the last bit of the network prefix, these elements will be taken together as a pair ⟨ A, B ⟩ where A is formed by taking the IP address of one of the elements and making the last bit of the network prefix (in binary notation) 0. B is the subnet mask of one of the elements with the last 1-bit (in binary notation) made 0.

   - e.g. ⟨ 192.168.8.0, 255.255.255.0 ⟩ and ⟨ 192.168.9.0, 255.255.255.0 ⟩ will be replaced with ⟨ 192.168.8.0, 255.255.254.0 ⟩

4 Repeat the previous step untill the current element cannot be taken together with the previous or the next one.

5 If there are more elements to be inserted, go to step 2, otherwise stop.

The identical algorithm is possible on the redundancy check table.

## 2. Simulation Study

**Simulation Parameters.** For our simulations we will use the nsclick [3] simulation platform. Nsclick embeds the Click Modular Router Platform [4] into the ns-2 [1] simulator allowing us to run actual routing protocols developed for the click platform inside a simulation. We based our work on a click based implementation of the OLSR protocol. The parameters used for OLSR are the default parameters described in [6].

**Simulated Scenario.** The scenario consists of an ad hoc network that is split into 4 clusters with 16 nodes each, one of which is a backbone node. The only way that a node can communicate with a node that is in another cluster is through the overlay network. Except for the backbone nodes, the nodes are placed randomly inside the area of the cluster they belong to. The backbone nodes are placed in the center of the area. An example is given in Fig. 5: the backbone nodes are shown in grey. The backbone nodes are configured with two 802.11 interfaces, that operate on different channels. The interface that is

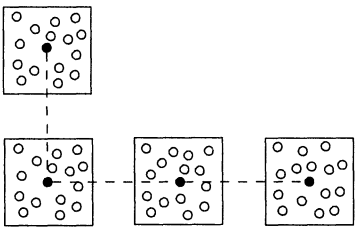

*Figure 5.* Simulated Scenario

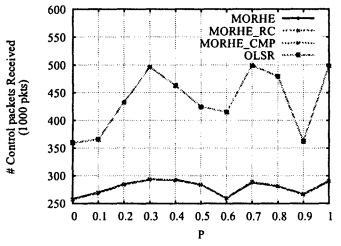


*Figure 6.* Routing signalling overhead (# packets received)

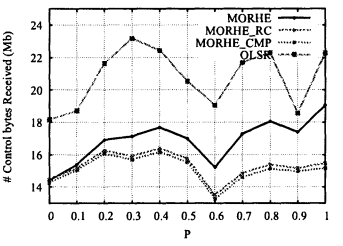


*Figure 7.* Routing signalling overhead (# bytes received)

connected to the overlay network has a range of 300m, while the interface on the subnet has a much smaller coverage (100m). The nodes in a cluster are randomly distributed on a 200m by 200m surface.

Each cluster is assigned 256 consecutive addresses: 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24 respectively. The backbone nodes are given the 192.168.x.253 address for the interface on the subnet and the 192.168.x.254 address for the interface on the backbone. The nodes are addressed upwards from 192.168.x.1.

For this scenario we define a parameter $P$ as the probablity that a node is visiting another cluster during the simulation run. P is varied from 0.0 to 1.0 in steps of 0.1. The nodes that are visiting another cluster are uniformly distributed among the remaing clusters. Using this scenario we can investigate the behaviour of the MORHE protocol in scenarios with a varying amount of visiting nodes in the clusters.

10 pairs of nodes initiate a unidirectional call, with each call staggered by 30s. The calls are CBR connections sending packets of size 500 bytes at a rate of 16 kbps.

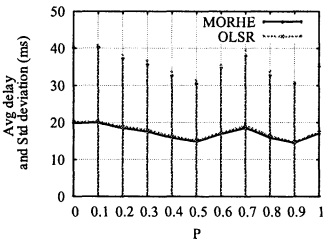


*Figure 8.* Average end-to-end delay

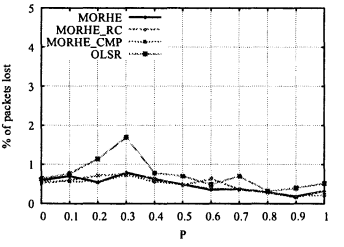


*Figure 9.* % packets lost

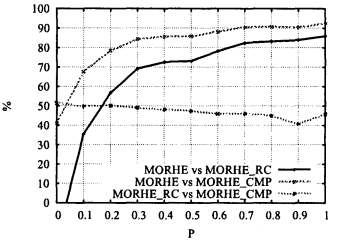


*Figure 10.* Reducing the overhead with redundancy check and compacting

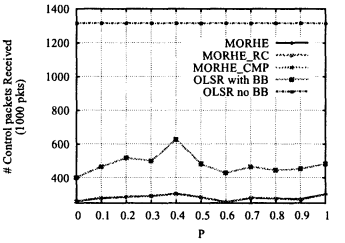


*Figure 11.* Reducing the overhead with redundancy check and compacting

## 3. Simulation Results

We will now compare the results of:

- OLSR using Multiple Interface Description (MID) support: MID should be supported by any OLSR implementation that wishes to comply with RFC3626 (In the graphs we refer to this as OLSR). This approach will actually make sure that one big ad hoc network is created.
- MORHE without redundancy check and IP address compating
- MORHE with redundancy check (In the graphs we refer to this as MORHE_RC).
- MORHE with redundancy check and IP address compacting (In the graphs we refer to this as MORHE_CMP).

We will start by looking at the total amount of overhead measured as the amount of routing signalling packets and bytes received during the simulation run. The results are shown in Figs. 6 and 7. As you can see in Fig. 6 MORHE offers a reduction in the amount of signalling packets received of about 40%

independently of the amount of nodes visiting a foreign cluster. Neither the redundancy check, nor the compacting have an effect on this result, since they only reduce the size of the HNA packets.
We can observe the same behaviour for the amount of bytes sent when there are less than 60% of the nodes visiting another cluster.(Fig. 7). However in scenarios where 60% or more nodes are visiting another cluster the overhead in bytes received increases and comes close to that of OLSR. This can be explained by the increase in the size of the HNA packets when there are more visiting nodes inside a cluster. Fig. 10 shows exactly how much using the redundancy check and the IP address compacting improves the MORHE protocol in terms of HNA overhead. It is clear that the redundancy check removes much of the overhead when many nodes are visiting another subnet. If both redundancy check and IP address compacting are active the overhead in bytes caused by the HNA packets is reduced by almost 90% if P is large. In fact by choosing the address ranges for the subnets in such a way that they can be compacted when advertised on the subnet - 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 and 192.168.3.0/24 compact to 192.168.0.0/22 - the HNA overhead, compared to using only redundancy checks, is reduced by almost 40% to 50% depending on the value of P. If one of the subnets were to become unreachable at some point, the overhead would increase since MORHE would not be able to compact the addresses into one subnet anymore, however connectivity among the remaining subnets is still guaranteed.
Fig. 9 shows that MORHE and OLSR have similar results when it comes to packet loss. Almost no packets are lost. 0ne can observe a slight improvement which is caused by the medium being less congested by broadcast packets, which is a direct consequence of the fact that MORHE reduces the overhead in the network. The average end-to-end delay in the network is shown in Fig. 8. For clarity only MORHE without extensions and OLSR are displayed and in terms of end-to-end delay there is no significant performance difference between MORHE and OLSR.

## 4. Conclusion

In this paper we have presented a transparent multi-level routing protocol based on OLSR. Our approach takes into account the heterogenous nature of the nodes in an ad hoc network and builds an ad hoc infrastructure that improves the scalability of the OLSR protocol. MORHE was developed in such a way that any OLSR compliant node can join a MORHE network and roam the different ad hoc clusters without any problems.
We have described how this protocol can be easily integrated into a working OLSR implementation and have implemented the protocol using an OLSR implementation for the Click Modular Router Project. Using the ns-2 simulator

and the nsclick extension we simulated several scenarios allowing us to compare OLSR with the MORHE protocol. The following conclusions can be made: Proactive ad hoc routing protocols need a large amount of signalling overhead to work and do not scale to large networks. MORHE avoids much of this overhead by dividing the network into smaller clusters and building an ad hoc infrastructure. Although the MORHE protocol was not primarily designed for scenarios where many nodes are visiting a foreign subnet the protocol behaves well even in these scenarios. By managing the address space of the ad hoc network, schemes such as the redundancy check and IP address compacting can be used to decrease the size of the signalling packets, while leaving the protocol behaviour unchanged.

## 5. Future Work

The focus in this paper was on scenarios with a static overlay network and non-overlapping clusters. We intend to further investigate the consequences of nodes which roam between overlapping ad hoc clusters, as well as the effects of mobile clusters. Scenarios based on group mobility models will be investigated to measure the performance of the MORHE protocol when the backbone network is more dynamic. Support for fast handovers between clusters should be a further research topic.

## References

[1] The Network Simulator ns-2

[2] OOLSR: Projet Hipercom, INRIA.

[3] Michael Neufeld, Ashish Jain, Dirk Grunwald, Nsclick: bridging network simulation and deployment, In *Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems, Atlanta, Georgia, USA*, 2002

[4] Eddie Kohler, The Click modular router, November 2000

[5] Andreas Tønnesen, Implementing and extending the Optimized Link State Routing protocol, Master thesis, UniK, November 2004

[6] P. Jacquet, T. Clausen, RFC 3626: Optimized Link State Routing Protocol (OLSR), Oct 2003.

[7] C. Perkins, E. Belding-Royer, S. Das, RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003.

[8] David B., Johnson David A., Yih-Chun Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), July 2004.

[9] G. Pei, M. Gerla, Mobility Management in Hierarchical Multi-hop Mobile Wireless Networks, In *Proceedings of IEEE ICCCN'99, Boston, MA, pp. 324-329*, 1999

[10] C. Perkins, RFC 3344: IP Mobility Support for IPv4, August 2002

[11] Y. Ge, L. Lamont, L. Villasenor, Improving Scalability of Heterogeneous Wireless Networks with Hierarchical OLSR, August 2004