IETF MANET   C. Adjih
Internet-Draft   S. Boudjit
Expires: January 19, 2006   P. Jacquet
                                                              A. Laouiti
                                                          P. Muhlethaler
                                            INRIA Rocquencourt, France
                                                                 Pr. Mase
                                           Information and Communication
                                        Network Lab., Niigata University
                                                           July 18, 2005

Address autoconfiguration in Optimized Link State Routing Protocol
draft-laouiti-manet-olsr-address-autoconf-01.txt

Status of this Memo

Copyright Notice

Abstract

Several MANET routing protocols have been recently promoted to experimental RFCs. However, autoconfiguration of MANET networks is still an unsettled area. This document proposes a protocol for autoconfiguration for both IPv4 or IPv6. Its corner stone is an conflict-detection algorithm. It aims at conceptual simplicity: essentially, each node periodically sends its addresses and an identifier. Conflicts are detected as identifier mismatches. This protocol might be used with any MANET protocol, although it naturally suits the OLSR routing protocol (on which we focus), with a light increase of control message overhead.

Table of Contents

1. Introduction

Mobile Ad hoc NETworks (MANETs) are infrastructure-free, highly dynamic wireless networks, where central administration or configuration by the user is very difficult. In hardwired networks nodes usually rely on a centralized server and use a dynamic host configuration protocol, like DHCP[10], to acquire an IP address. Such a solution cannot be deployed in MANETs due to the unavailability of any centralized DHCP server. For small scale MANETs, it may be possible to allocate free IP addresses manually.

However, the procedure becomes impractical for a large-scale or open systems where mobile nodes are free to join and leave. Most of the autoconfiguration algorithms proposed for ad hoc networks are independent of the routing protocols and therefore, generate a significant overhead. Using the genuine optimization of the underlying routing protocol can significantly reduce the autoconfiguration overhead.

Because traditional IP solutions to autoconfiguration cannot be used as is on MANET networks, a MANET-AUTOCONF effort was set with three initial goals, which include the two following: an "IPv6 stateless autoconfiguration mechanism", and a "mechanism promoting address uniqueness in the situation where different ad hoc networks merge". This document proposes a protocol that addresses these two issues. The third one, "stateful address autoconfiguration mechanism", might be addressed by derivative of the method of the draft.

This comprehensive scheme centers on one control message and one mechanism, which ensure at the same time conflict avoidance in the 2-hop neighborhood of each node, and conflict avoidance in the entire network. This algorithm operates on multiple interfaces and is shown to work in any case of multiple conflicts, especially during network mergers.

1.1 Changes

Major changes from version 00 to version 01

- Changes in the structure of the document, and important editorial changes.

- DAD and MPR flooding is ensured in case of multiple interfaces.

- The second (now third) approach does not modify the basic OLSR HELLO message format anymore.

- Another approach has been added.

1.2 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [14].

For the OLSR description and terminology, the reader should refer to the OLSR RFC [12].

2. Problem Statement

The problem solved by the proposed autoconfiguration protocol is the autoconfiguration in MANET networks. The reader is referred to the autoconfiguration survey in [1], for the general problem statement, and a survey of several proposed solutions.

Considering the connectivity scenarios listed in [3], our prime objective is the autoconfiguration of isolated MANETs, but extending it to MANETs intermittently connected, or to connected MANETs is intended and already proposed ([5]).

More importantly, our main goal is promoting configured address uniqueness in the situation where different ad hoc networks merge. Precisely, the networks may be isolated, may become partitioned, and may merge, ... Under these hypothesis, the only way to ensure address uniqueness is that, in the critical case of a merge of two networks, the information about the all the addresses of the first network is compared to the information about all the addresses of the second network. That can be done in direct or indirect ways, with centralized or distributed means (and anything in between).

3. Overview of the Method

With respect to the problem statement, the method is build around a conceptually simple means of ensuring that information about all addresses is checked: each node sends its lists of addresses periodically, and a node identifier. Hence the approach is fully distributed, and introduces an unique new kind of message. It deliberately favors simplicity at the expense of bandwidth efficiency. At least in the case of the OLSR protocol, this is not an issue.

More precisely, our proposed auto-configuration algorithm is structured around three building blocks:

1. Address assignment: an IP address is selected by the arriving node.

2. Duplicate Address Detection (Conflict Detection): each node checks that there is not another node with the same address.

3. Conflict resolution: when a node detects that another node is using the same address, it will select a new address.

This method may also integrate a number of optional elements, namely:

1. Route contamination avoidance and gradual entry (borrowed from [4]).

2. Interface with gateways (see also [5]).

3. Passive duplicate detection methods.

The following sections give an overview of each part of the protocol.

3.1 Address Assignment

Numerous schemes can be used to select an IP address. For instance, the node can perform a random selection; another technique is that each node may advertise a set of addresses that (it believes) are not used, for potential newcomers as in [2]. Another method, more direct, is that a neighbor node selects this IP address for the arriving node by control message exchanges [14].

Because it is assumed that the MANET network may be isolated, a default method is to choose an address at random inside a given subnet with MANET_PREFIX: the pool of addresses could be for local use only. For example, it could be reserved by the IANA authority for local MANET forwarding (i.e, those addresses must not be

forwarded outside the MANET network, nor reached from outside). Choice may be more subtle, see Section 3.5. Additional addresses may be added, see Section 3.4.

3.2 Duplicate Address Detection

As highlighted previously, the DAD algorithm uses a single special control message to perform conflict detection.

This control packet includes one identifier and all the addresses of the node. This message is periodically transmitted to the entire network. The identifier of each node is assumed unique (with sufficient probability). If a node receives a message with a different identifier than its own, an address duplication is detected and the node selects a new address.

This control message is called MAD, for "Multiple Address Declaration". It is an extension of MID messages for OLSR and it also advertises all acquired addresses of the node. Because it is the central part of this method, it is described in the Section 4.

3.3 Conflict Resolution

When two nodes A1 and A2 are configured with the same IP address and assuming that there is no packet loss, at least one of these two nodes will receive the MAD message of the other node. Thus the nodes where the conflict lies are bound to discover the conflict, and can resolve it by changing addresses. Since a conflicting node knows via the reception of the MAD control messages the already assigned addresses, the new address must be selected at random among the addresses that are believed to be free.

3.4 Optional: MANET connected to an external network

When a MANET is connected to an external network, in case of IPv6, the node may receive IPv6 router advertisement messages. The intent of MAD messages was to advertise all the addresses of a node, including newly formed global addresses when such an IPv6 router advertisement is received (diffused by a manet multicast), see [14]. A better and more complete protocol to achieve the same goals (and more) is described in [5], and it can be adapted directly to the method presented here.

3.5 Optional: Routing Table Contamination Avoidance

In the case that node has just changed its address, an useful technique is to perform some routing table contamination avoidance (pioneered in [4]). It consists into letting a node entering

gradually in the network, going through several states: at first it is only recognized by its immediate neighbors, but not advertised, and not used for routing. This gives the node opportunity to detect (passively) an address conflict with an existing node, and change its address.

3.6 Optional: Passive Duplicate Detection

When the OLSR routing protocol is used (version 1 or version 2), it is possible to use a derivative of passive detection techniques found in NOA-OLSR [4] and [2], and pioneered by PACMAN [9]: the topological information diffused by the OLSR routing protocol is sufficient to detect address conflict. The address conflicts are essentially detected in two ways, as analyzed in [9]: inconsistent topological information (essentially, a node discovers that a control message incorrectly advertises that it has a link), and inconsistent message originators (a node discovers that it is credited for originating a message, which actually, it did not transmit).

Using passive techniques, one still need to ensure that control messages are propagated properly. Especially in the case of OLSR with multiple interfaces (but also in the cases given in the figures of [9]), it is necessary to ensure that MPR selection is done properly: we propose that MPR selection is ensured by the propagation of MAD messages at only a distance up to two hops from the originator, (4 hops when conflict is detected, as the algorithms proposed here do), which is enough to guarantee sufficient resolution of short range conflict to permit proper MPR selection.

Hence MAD messages are sent only in a limited local area, to guarantee proper MPR selection, at limited cost; and longer range conflicts are detected by passive methods.

For completing all theoretical conflict cases, and in order to accelerate the detection by passive methods, in the even rarer case when the topology is symmetric, and nodes in conflict have identical message sequence numbers, we suggest that one bit somewhere in the message may be set randomly in the message (or based on node identifier): this gives an additional 50% chance of resolving the conflict at each generated message. (such a bit that might be set, is the last bit of the message sequence number).

4. Duplicate Address Detection Algorithms

4.1 Overview of the Different Solutions

In order to detect address conflicts, each node diffuses periodically a special message that we call a MAD for ``Multiple Address Declaration'' to the entire network[14]. This control packet includes the node addresses and the node identifier.

The node identifier is a sequence of bits of fixed length L which is randomly generated. Hence we are using the standard idea that the probability of two nodes having the same node identifier is low, and the probability of at least one address collision with N nodes, which is the well known ``birthday problem'', can be set arbitrarily low by choosing a large enough value of L (eight bytes are enough to code the random identifier if we consider a network with a maximum of 10000 nodes [13] [16]). The MAD message format is depicted in the Appendix A

A node detects an address conflict when it receives an MAD message having the same address as its own, but with a different identifier. These situations may happen during network mergers. Actually other nodes will detect the conflict. These nodes could announce the conflict using a special control message. However this approach may induce broadcast storm since many nodes may announce the conflict and special care must be taken to avoid this effect. For that reason we do not recommend this way. An efficient manner to notify the address duplication to the nodes in conflict, consists in allowing the MAD packets to reach all the nodes in the network.

Depending on which routing protocol is actually used, the MAD messages may be optimized in several kind of ways. Several cases may be identified:

o OLSR [12]

o OLSRv2 [6]

o A protocol with support for an MPR-based flooding (such as [7] and [8])

o A protocol without an MPR-based optimization

In the last case, if the protocol has no MPR-based optimization, the sending of MAD messages is done with the default of using pure flooding. Additionally, whenever OLSR is not used, the MAD messages must be encapsulated in proper packets.

Otherwise: when MPR-based flooding is present, to save channel bandwidth the MAD packets should be broadcasted using this MPR-based flooding (either as a MPR-CDS or a genuine MPR flooding). Then, of course, MAD messages are used in order to resolve conflicts, but conflicts themselves can lead to improper MPR selection. Hence a few theoretical situations result some conflicts may not be resolved. This is addressed by special relaying rules for MPR-flooding, and three different approaches:

- In first one, designed for OLSR, we suggest to ignore the MPR mechanism for MAD relaying in the one-hop neighborhood and instead, the MAD messages are repeated by all neighbors (one-hop pure flooding). Moreover special rules for address conversion are introduced when using the content of MAD messages. One-hop pure flooding is sufficient for OLSR, hence the overhead is limited.

- The second one focuses on protocols different from OLSRv1: for MPR-based flooding protocols, and for OLSRv2, it is necessary that MAD messages are repeated by neighbors and also 2-hop neighbors. This method is a derivative of the previous algorithm, and is slightly more expensive, but more general.

- In the third one, MAD messages are also relayed by one-hop neighbors, as in the first approach. Moreover, we also modify the MPR election algorithm of OLSR. The calculation will be based on node identifiers to guarantee the uniqueness of the direct neighbors and the 2-hop neighbors. This leads to a correct MPR set, hence, ensures that the MAD messages reach all the nodes.

4.2 First approach

This approach is based on new rules for MPR flooding for MAD message (details are given in Appendix B). The proof of correctness of this algorithm is given in [15] (including cases with multiple interfaces and multiple conflicts). The two rules are:

4.2.1 Rule 1

When a node X receives a MAD message and if node X has a symmetric or asymmetric link with a node Y with the same main address as the one contained in the MAD message, then node X relays this MAD message. When relaying the MAD message the Hop-Count field is set to 1.

4.2.2 Rule 2

When a node X receives a HELLO message from a node Y, this HELLO contains interface addresses of 2-hop neighbors of X(1-hop neighbors

of Y). To convert such addresses into main addresses the node X uses MAD messages that are relayed by Y. The rule 2 will actually avoid inconsistent main address conversions for 2-hop neighbors. This is essential in the case of nodes with multiple interfaces.

4.3 Second approach

In this approach, it is assumed that the rule 2 of the first approach can no longer be applied, because, for instance, the protocol does not use or has not MID/MAD messages. In this case, the rule 2 is replaced the rule 2B, and the details are easily derived from Appendix B:

4.3.1 Rule 2B

When a node X receives a MAD message, which it has not already retransmitted, with a hop count field equal to one, it relays it. The rule 2B will actually make all the 2-hop neighbors retransmit the message.

4.4 Third approach

An issue with this first approach is that the conflicts at 2-hop must be resolved before one can be sure that the MAD messages are successfully transmitted within the entire network. An ideal property would be that the MAD messages reach all the nodes in the network irrespectively of potential address duplications. This property can be achieved if the MPR flooding continues to work in presence of address duplication. A solution is then to base the selection of MPR not on addresses but on node identifiers. With the assumption that node identifiers are globally unique in the network, one can be sure that there will not be identifier duplications at two hops of a given node and thus the selection of MPRs will be correct. This solution can be simply implemented, the selection of the MPRs must follow the principle defined in the OLSR protocol except that the base for the selection must be the node identifiers i.e. the 2-hop coverage must be considered not on the addresses but on the node identifiers.

This is achieved in this section by providing an alternative to the second rule. The Rule 2 is replaced by a Rule 2C:

4.4.1 Rule 2C

The MPR calculation is modified, by using node identifiers. Each address is converted to a node identifier (using a method described later): as a result the node computing its MPR set, has its 1-hop and 2-hop topology represented by links between node identifiers.

Details about applying rule 2C in practice are found in Appendix C.

5. IANA Considerations

One new type of control message is defined in this protocol, for MAD messages.

6. Security Considerations

This memo does not specify any security considerations.

Appendix A. MAD Message Format

An example of MAD message format is depicted in the following figure

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                   Originator node identifier                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     OLSR Interface Address                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     OLSR Interface Address                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              ...                              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1

An alternative is to prepend an "Originator node identifier" OLSR message, to MeID messages

Appendix B. DAD-MPR flooding algorithm

Let us recall the assumptions here. Each node A periodically sends a message M including:

- The originator address of A, Orig_A, in the OLSR message header.
- The list of interface addresses of node A in the message it self.
- The message sequence number, mssn, in the OLSR message header.
- The node identifier ID_A (a string of bits) in the message itself.

The message is propagated by MPR flooding to the other nodes ; but for DAD-MPR Flooding, the duplicate table of OLSR is modified, so that it also includes the node identifier list in the duplicate tuple. That is, a duplicate tuple, includes the following information:

- The originator address (as in OLSR standard duplicate table).
- The message sequence number (as in OLSR standard duplicate table).
- The list of node identifiers.
- The list of interface addresses on which the MAD message was received.

The detailed algorithm for DAD-MPR Flooding is the following:

- When a node B receives a MAD message M on its interface B(i) from node C with originator Orig_A, with message sequence number mssn, and with node identifier ID_A, it performs the following tasks:

  1. If a duplicate tuple exists with the same originator Orig_A, the same message sequence number mssn, the ID_A is in the list of node identifiers, and the interface address of B(i) is in the list of interface addresses on which the message has already been received, Then, the message is ignored (it has already been processed). The algorithm stops here.

  2. Else one of the following situations occurs :

     1. A duplicate tuple exists with the same originator Orig_A, the same message sequence number mssn, ID_A is in the list of node identifiers, but the address of B(i) is not in the list of interface addresses on which the MAD message has already been received: then, the message must be

processed. The address of B(i) is added to the list of interface addresses on which the message has already been received.

2. A duplicate tuple exists with the same originator Orig_A and the same message sequence number, but ID_A is not in the list of node identifiers: then, a conflict is detected (address Orig_A is duplicated). ID_A is added to the list of node identifiers.

3. A duplicate tuple exists with the same originator Orig_A, but with a different message sequence number and ID_A is not in the list of node identifiers: then, a conflict is detected (address Orig_A is duplicated). A duplicate tuple is created with the originator address, message sequence number and list of node identifiers containing only ID_A.

4. No duplicate tuple exists. A new one is created with the originator address Orig_A, message sequence number mssn, list of interface addresses on which the MAD message has already been received containing only the address of B(i), and list of node identifiers containing only ID_A.

3. The MAD messages should be relayed if one or more of the following rules are met:

    1. C had chosen this current receiving node, B, as an MPR.

    2. Node B has a symmetric or asymmetric link with a node Y with the same main address as the one contained in the MAD message M. In such a case, the Hop-Count field is set to 1 before message M retransmission.

- When a node X receives a HELLO message from a node Y, this HELLO contains interface addresses of 2-hop neighbors of X. The node X uses MAD messages relayed by Y to convert such addresses into main addresses.

Appendix C. Precisions for third approach

The goal is to obtain the one hop neighborhood and the two-hop neighborhood with node identifiers in place of addresses.

Converting main addresses of neighbor to node identifiers is easily done: when receiving the MAD messages from neighbors, the main address can be identified to be the address of a neighbor, and the node identifier is given. Hence the node may record the information mapping main addresses of neighbors to their identifiers.

Converting main addresses of two-hop neighbors to node identifiers is less direct: however the information obtained thanks to the fact that MAD messages from two-hop neighbors are always retransmitted by one-hop neighbors. Such MAD messages are identified by the fact that they arrive with a hop count field (corrected by Rule 1 is necessary), equal to 1 ; in the following, they are called two-hop MAD messages. The receiver node can thus maintain the information Two-Hop Identifier Table: (neighbor address, two-hop neighbor addresses list, two-hop neighbor identifier) in or in addition to, the MID/MAD information base. Now taking advantage of the fact that conflicts at distance 1 to 3 are resolved anyway by Rule 1, it is know that one neighbor will retransmit neighbor MAD message (two-hop MAD messages for the receiver) that have all different addresses: otherwise there would be a 2-hop conflict, necessarily resolved. Thus the mapping deduced from the Two-Hop Identifier Table, (neighbor main address, two-hop neighbor main address) -->two-hop neighbor identifier is unique (and corresponds to reality).

Note also, that the information containted in the two-hop identifier table, should be also used for processing Hello messages (converting interface addresses to main addresses) so that the two-hop neighbor main address in the two-hop tuple is the actual main address.

7. References

7.1 Normative References

7.2 Informative References

[1] Bernardos, C. and M. Calderon, "Survey of IP address autoconfiguration mechanisms for MANETs", draft-bernardos-manet-autoconf-survey-00 (work in progress), July 2005.

[2] Clausen, T. and E. Baccelli, "Simple MANET Address Autoconfiguration", draft-clausen-manet-address-autoconf-00 (work in progress), February 2005.

[3] Ruffino, S., "Connectivity Scenarios for MANET", draft-ruffino-conn-scenarios-00 (work in progress), February 2005.

[4] Mase, K. and C. Adjih, "No Overhead Autoconfiguration OLSR", draft-mase-manet-autoconf-noaolsr-00 (work in progress), May 2005.

[5] Ruffino, S. and P. Stupar, "Automatic configuration of IPv6 addresses for nodes in a MANET with multiple gateways", draft-ruffino-manet-autoconf-multigw-00 (work in progress), June 2005.

[6] Clausen, T., "The Optimized Link-State Routing Protocol version 2", draft-clausen-manet-olsrv2-00 (work in progress), July 2005.

[7] Perkins, C., "Multicast With Minimal Congestion Using Connected Dominating Sets", draft-perkins-manet-smurf-00 (work in progress), July 2005.

[8] Macker, J., "Simplified Multicast Forwarding for MANET", draft-ietf-manet-smf-00 (work in progress), July 2005.

[9] Weniger, K., "Passive Duplicate Address Detection in Mobile Ad hoc Networks", March 2003.

[10] "R. Droms, Ed.,J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, 'Dynamic Host Configuration Protocol for IPv6 (DHCPv6)', IETF RFC 3315, July 2003".

[11] "S. Thomson, T. Narten, 'IPv6 Stateless Address Autoconfiguration', IETF RFC 2462, December 1998".

[12] "T. Clausen, Ed., P. Jacquet, Ed., 'Optimized Link State Routing Protocol', Request for Comments (Experimental) 3626, Internet Engineering Task Force, October 2003".

[13] "S.Boudjit, A. Laouiti, P. Muhlethaler, C. Adjih, 'Duplicate address detection and autoconfiguration in OLSR', INRIA Research Report-5475, Jan 2005".

[14] "A. Laouiti, S. Boudjit, P. Minet and C. Adjih, 'OLSR for IPv6 networks', Proceedings of Med-Hoc-Net, June 2004,Bodrum, Turkey".

[15] "C. Adjih, S.Boudjit, P. Jacquet, A. Laouiti, P. Muhlethaler, 'An Advanced Configuration and Duplicate Address Detection mechanism for a multi-interface OLSR Network', INRIA Research Report, Jul 2005".

[16] "S. Boudjit, C. Adjih, A. Laouiti, P. Muhlethaler,'Duplicate address detection and autoconfiguration in OLSR', Proceedings of IEEE SAWN 2005, May 2005, Maryland, USA".

Authors' Addresses

Cedric Adjih
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5215
Email: cedric.adjih@inria.fr

Saadi Boudjit
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5039
Email: saadi.boudjit@inria.fr

Philippe Jacquet
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5263
Email: philippe.jacquet@inria.fr

Anis Laouiti
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5088
Email: anis.laouiti@inria.fr

Paul Muhlethaler
INRIA Rocquencourt, France
Project HIPERCOM
Domaine de Voluceau -Rocquencourt
BP 105
Le Chesnay 78153 cedex
France

Phone: +33 1 3963 5278
Email: paul.muhlethaler@inria.fr

Pr. Kenichi Mase
Information and Communication Network Lab.,Niigata University
Niigata University
Niigata 950-2181,
Japan

Phone: +81 25 262 7446
Email: mase@ie.niigata-u.ac.jp
URI: http://www.net.ie.niigata-u.ac.jp/

Intellectual Property Statement

Disclaimer of Validity

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

Copyright Statement

Acknowledgment

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