Duplicate Address Detection in OLSR Networks

Thomas Heide Clausen, Emmanuel Baccelli, Julien Garnier
T.Clausen@computer.org, Emmanuel.Baccelli@inria.fr, Julien.Garnier@polytechnique.fr
PCRI – Pole Commun de Recherche en Informatique du plateau de Saclay,
CNRS, Ecole Polytechnique, INRIA, Universite Paris Sud
Laboratoire d'Informatique, Ecole Polytechnique, 91128 Paliseau, FRANCE

***Abstract*—**

Commonly, duplicate address detection is performed when configuring network interfaces in order to ensure that unique addresses are assigned to each interface in the network. Such mechanisms commonly operate with the premises that a node "intelligently" selects an address which it supposes to be unique, followed by a duplicate address detection cycle, through which it verifies that no other active interfaces on the same network has been or is in the process of being configured with the same address. Even assuming that such a mechanism is present in a MANET, allowing MANET nodes to initially configure their interfaces with addresses unique within the network, additional complications arise: two or more MANETs may merge to form a single network, and a formerly connected MANET may partition. Thus, unless it is ensured that all MANET interfaces are assigned globally unique addresses, addressing conflicts may at any point – not just during initial network configuration.

In this paper, we investigate the task of performing duplicate address detection when otherwise independent OLSR networks merge. We benefit from the information already exchanged by OLSR, and identify a number of mechanisms through which a node may detect a conflict between the address assigned to one of its interfaces, and an address assigned to an interface on another node. The mechanisms proposed are, thus, entirely passive, creating no additional information exchange on the network.

I. Introduction

A Mobile Ad hoc NETwork (MANET) is a collection of independant nodes, communicating over a wireless medium, whereby they form a multi-hop connected network with no assumptions of an a-priori network infrastructure. Due to the nature of the communications interfaces, protocols governing the connectivity and operation of such MANETs must be designed such that they restrict the amount of information exchange required between the nodes – in other words: the available bandwidth within the MANET should as far as possible be preserved for data communication. Thus, routing protocols for MANETs should be designed with minimal control-traffic overhead as a key criteria. An additional constraint on MANET routing protocols is, that nodes in a MANET may move freely, thus dynamically changing the topology of the network. As such, a MANET routing protocol must also satisfy the requirement of being able to respond to potentially frequently changing topologies – while still aim to minimize the information exchange.

A. The Optimized Link-State Routing Protocol

A premiere routing protocols for MANETs is the Optimized Link-State Routing protocol, OLSR [3]. Designed for low-bandwidth high-dynamics networks, OLSR employs periodic optimized flooding of link-state information using multi-point relays [5]. Additionally, OLSR employs partial link-state information, ensuring that all destinations, but only a subset of links sufficient to ensure shortest-paths, are known by all nodes in the OLSR network.

In this section, we will expose attributes of OLSR, relevant for performing duplicate address detection in case two or more OLSR-networks merge. Thus, the three relevant messages, exchanged between OLSR nodes, are briefly described below. The information exchanged in these messages will be employed in section II.

1) HELLO messages: HELLO messages are employed by OLSR to detect local link-state changes and perform bidirectionally checks of links, and are thus exchanged between neighbor nodes only. A node will, periodically, generate and transmit a HELLO message, containing three lists: a set of neighbor node interfaces which have been "heard", but where bidirectionally hasn't yet been confirmed (status: ASYM), a set of neighbor node interfaces with which bidirectionally has been confirmed (status: SYM) and a list of neighbor nodes which are designated as multi-point relays (status: MPR). The MPR designation implies (for the purpose of this paper) two things: that the link to an MPR designated node is symmetric (*i.e.* the MPR nodes form a subset of the SYM nodes), and that the MPR node will generate and diffuse link-state information advertising the link

between themselves (the MPR) and the node which selected it (the MPR selector).

2) TC messages: TC messages in OLSR are the functional equivalent of OSPF LSA-messages in that they diffuse link-state information throughout the network. Thus, an MPR node will periodically generate a TC-message, in which it lists its MPR selectors. Since all nodes will select MPRs from among their symmetric neighbors, all nodes – as well as the links nodes and their MPRs – will be diffused throughout the net. Since TC messages are diffused periodically, a sequence-number is included in each TC message, allowing recipients to determine which of two TCs from the same node is the most recent.

3) MID messages: In OLSR networks, nodes may have multiple interfaces, each with a distinct address, participating in the MANET. Such nodes with multiple interfaces will, periodically, generate MID messages and diffuse to the entire network, listing the set of interface addresses which participate in the OLSR network.

B. Duplicate Address Detection in OLSR networks

Typically, addresses are assigned in ways which (try to) ensure uniqueness: no two interfaces within the same network should carry the same address. In traditional, wired, networks this has been ensured by state-full centralized servers [4] or through stateless autoconfiguration [6] – the former assuming that a centralized server is always present and reachable in the network, the latter that all interfaces in the network share a broadcast/multicast link, over which they are reachable and able to participate in a distributed address assignment algorithm.

In an OLSR network, these approaches fail: no centralized entity can be assumed – indeed, is contrary to the definition of a manet – and even if one was present, mobility and the potentially fragile nature of radio-links would imply that this centralized entity could be reached by all interfaces at all times. Similarly, the requirement that interfaces share a broadcast/multicast link fails: an OLSR promotes multi-hop communication through the deployment of a routing protocol among the nodes – which depends on interfaces already having uniquely configured address in order to operate.

Several proposals for initial address configuration exist[2], [1] etc., providing different ways for nodes within an OLSR network to initially acquire unique addresses. A common thing for these proposals is, however, that the nodes acquiring address must form a multi-hop connected network.

However with OLSR networks being inherently mobile, network partitioning, where a set of nodes loose connectivity from the rest of the nodes, and network merger of formerly independent or partitioned networks may occur. In the situation where two network merge, there is a non-zero probability that nodes in the two networks may have configured their interfaces with the same addresses – or more precisely, there's no guarantee that network interfaces across the two networks have unique addresses. Thus, as a supplement to a mechanism ensuring initial uniqueness of interface addresses, a mechanism is required to ensure that interface address uniqueness is preserved in the face of network mergers.

The purpose of this paper is to present ways in which an OLSR node can detect address conflicts in the case where already configured OLSR networks merge. More specifically, we present ways in which a node can detect if an address, currently assigned to one of its own interfaces, is concurrently used on another interface in the network. We note, that the approaches taken for detection of duplicate addresses is completely passive: no additional information exchange is required between the nodes – all that is required is for the nodes participating in the network to be running OLSR, and for the duplicate address detection mechanism to have access to the internal state of the OLSR routing daemon.

C. Paper Outline

The remainder of this paper is organized as follows: section II describes the different mechanisms, already present in OLSR, which can be exploited to provide the ability to detect when an address is concurrently assigned to more than one interface. Specifically, the methods do not incur any additional information exchange and, for all but one do not incur any additional storage requirements to the OLSR nodes. Once an address conflict is determined, conflict resolution can be performed, with the goal of removing any conflicting address assignments. Some additional considerations regarding the applicability and scope of the discussed mechanisms are included in section III. Section IV how resolving a conflict, once detected, can be performed, followed by section V, which concludes this paper.

II. Performing DAD in OLSR Networks

In this section, we present different mechanisms through which an OLSR node can detect if the address, currently assigned to one of its interfaces, is concurrently being used by an interface on another node. We note that none of the mechanisms presented here impose any additional information exchange between nodes beyond what is already performed by OLSR.

The duplicate address detection mechanisms are based on inspecting received OLSR control messages, as well as the receiving nodes state, to determine if an address on the receiving node is duplicated elsewhere in the network. More precisely, a node can inspect a received message to detect (i) if the message appears to have been sent from an interface the receiving node or (ii) if the message contains information about interfaces of the receiving node. In either of these cases, the information

contained in the received OLSR message is compared to the state recorded in the receiving node, allowing the receiving node to detect a potential duplicate of one of its addresses.

With this in mind, the following subsections will inspect the three OLSR message types described briefly in section I-A: HELLO, TC and MID-messages. In the illustrations in this section, nodes are identified by addresses ”A”, ”B”, ”C”, ... If an address is duplicated, two nodes will appear in the figure with the same address. The node being the center from which the mechanism is described is indicated by a double-circle.

A. HELLO: Mismatching neighborhood

A node which receives a HELLO message on one of its interfaces, where the HELLO message appears to come from the node itself, a potential address duplication may incur: since HELLO messages are never forwarded in OLSR, an OLSR node should not receive a copy of a HELLO message with any of its own interface-addresses as originators[1]. Should a node receive a HELLO message with one of its own interface addresses listed as originator, there’s a likely collision: two adjacent nodes have interfaces configured with the same address, as illustrated in figure 1. From the point of view of the leftmost node ”A” (indicated by a double-circle), this can be confirmed by inspecting the neighborhood being advertised in the HELLO message: the HELLO message will include nodes B and C as neighbors, whereas neither are neighbors of the node receiving the HELLO. Thus, it can be detected by the leftmost node in figure 1 that one of its interface addresses is also being used elsewhere in the network.

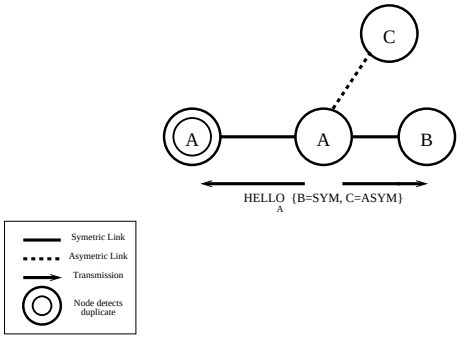

Fig. 1. Node A detects an address duplication as it receives a HELLO message with its own address listed as originator address.

B. HELLO: MPR Selection Abnormality

A second intuitive diagnostic on HELLO messages could be to consider MPR selection: an MPR node must be selected from among neighbors with which a symmetric link exist. Thus, a hypothesis could be that if the leftmost node ”A” on figure 2, which has a recorded asymmetric link with node ”B”, receives a HELLO from node B declaring it as MPR, then a conflict exists as indicated: a second node ”A’”, adjacent to ”B”, has the same address as ”A”.

This would, however, be a false conclusion. On establishment of the link between ”A” and ”B” node ”A” receives a HELLO from ”B”, bringing node ”A” to see the link to ”B” as ASYM. In the next HELLO from node ”A”, node ”B” will see its own address listed and conclude that the link is symmetric. Node ”B” may, then, select ”A” as MPR and include this selection in the next HELLO message. In this way, node ”A” will receive an MPR selection from a node with which it has only an asymmetric link, without this being an indication of address conflicts in the network.

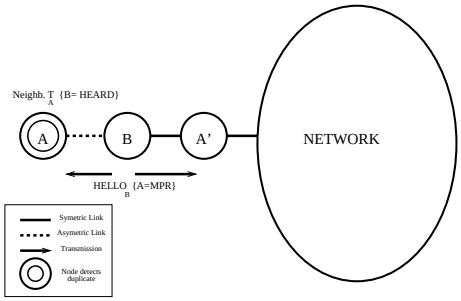

Fig. 2. Node A detects an address duplication as it receives a HELLO message with its own address listed as MPR from a neighbor B with which it has no symmetric link.

C. TC: Sequence Number Mismatch

If a node, ”A”, receives a TC message with the address of one of its own interfaces listed as originator address address and with a sequence number very different from the sequence number that node ”A” currently is using, this can be an indication that an interface of node ”A” is concurrently being assigned to another interface in the OLSR network. This situation is illustrated in figure 3

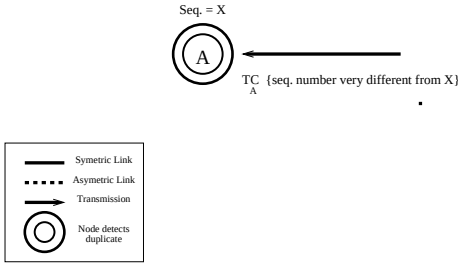

Fig. 3. Node A detects an address duplication as it receives a TC message with its own address listed as originator address and with a sequence number that is very different from the current sequence numbers it uses.

[1]This ignores the situation where a node has two radio interfaces running OLSR on the same channel.

D. TC: Link-State Mismatch

If a node, ”A”, receives a TC message, declaring the address of one of node ”A”’s interfaces as MPR selector, the originator of that TC-message must be a direct neighbor of node ”A”. Considering, however, the situation illustrated in figure 4: the rightmost node ”A” selects node ”C” as MPR, and thus node ”C” will advertise ”A” in its TC messages. From the point of view of the leftmost node ”A”, an address conflict will be detected thus: a TC will be received from node ”C”, advertising a link between node ”C” and node ”A”, yet in the leftmost node ”A” no such link exists.

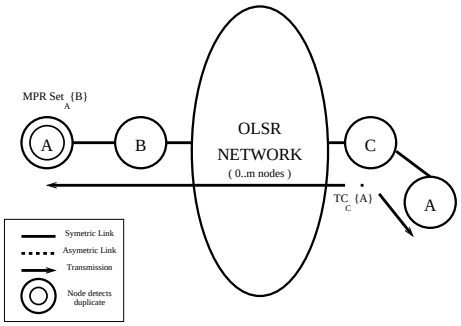

Fig. 4. Node A detects an address duplication as it receives a TC message with its own address listed as originator address and with listed addresses that are not in its neighborhood, or MPR selection.

E. MID: Interface Mismatch

The final message type in OLSR is MID messages, through which a node with multiple interfaces declares its interface configuration to the other nodes in the network. If a node, ”A”, receives a MID messages, in which the address of one of its own interfaces is listed, the remaining addresses listed in the MID must also belong to node ”A”. Alternatively, if a node, ”A”, receives an MID-message, containing one or more addresses, belonging to node ”A” but also listing addresses which do not belong to node ”A”, then at least one address is assigned to more than one node. This is illustrated in figure 5, in which one node has the addresses ”A”, ”A1” and ”A2, whereas the other node has the addresses ”A”, ”A3” and ”A4”.

III. Applicability and Scope

Passive mechanisms, such as those presented in this paper, are based on the monitoring of the control messages of the routing protocol. These aim at detecting anomalies in this traffic, that can hint to possible address collisions. However, this approach has a few shortcomings, both in terms of false alarms and in terms undetected duplications.

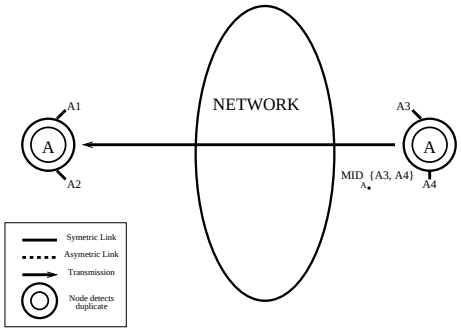

Fig. 5. Node A detects an address duplication as it receives an MID message with its own address listed as originator, and with listed addresses that are different from its own.

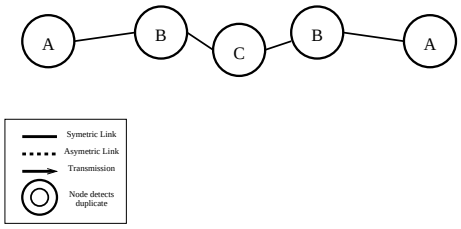

Fig. 6. A completely symmetric OLSR network, where tracking of control traffic fails to detect address duplication

In the rare case of a totally symmetric MANET, such as the one as depicted in figure 6, routing message monitoring may not be sufficient to detect the duplicate addresses. In figure 6, the duplicate nodes cannot detect the collision with each other since the routing messages produced by the left side of the network are identical to the routing messages produced by the right side of the network (because the topology is symmetric). Sequence number mismatch monitoring may help in this case, but it may also crash the network further, as such mismatches may invalidate the link state information with each TC transmission, alternatively from the right side and the left side of the network. Another example is with the sequence number mechanism. This technique is not completely reliable in order to detect duplicate addresses, as delayed delivery can cause an outdated control message that is received to be possibly wrongly interpreted as a case of address duplication. This category of false alarm is more likely to be caused by TC or MID messages rather than HELLO messages, as they feature only one hop scope, suppressing delays due to forwarding.

Such cases challenge the passive approach to DAD. Therefore other techniques maybe employed in addition to passive mechanisms in order to increase the reliability of the DAD. These techniques can be called active, or semi-passive, depending on how much additional overhead is produced by the mechanism.

Semi-passive techniques involve deeper analysis of the link state information traffic, such as tracking and pro-

cessing the history of such traffic, in order to prevent errors. However, these techniques come with much more processing and memory needs, a fact that must be carefully evaluated.

Active techniques involve sending specific DAD information or messages, in addition to the routing control overhead. For instance, flooding a neighbor solicitation message is part of such a technique. These can be more efficient than passive waiting, but they nevertheless come with greater overhead, a fact that must also be carefully evaluated.

IV. Resolving Duplicate Address Conflicts

The purpose of the mechanisms, described in this paper, is to detect when two or more interfaces in the network have been configured with the same address – that a duplicate address conflict exists in the network. The logical next-step to having detected this situation is to resolve it – to reconfigure nodes such that each interface participating in the OLSR network has a network-wide unique address.

Resolving a duplicate address conflict is, functionally, orthogonal to detecting a duplicate address conflict and, depending on the specificities of the network, different mechanisms can be employed. In this section, we briefly outline a few general approaches to resolving duplicate address conflict. The objective, however, remains to remove conflicting interfaces from the OLSR network, while disrupting the network operation as little as possible.

The simplest solution, once a duplicate address conflict is detected, is for a node to simply disable the local interface(s) which are conflicting. If these interfaces then wish to enter the network again, a new initial autoconfiguration cycle must be initiated. The advantage of this method is its simplicity and fact that no lengthy election procedure must be completed before duplicate address conflicts are resolved. The disadvantage is, that when a conflict arises, all conflicting interfaces are potentially disabled without consideration to traffic (or even necessity: when two interfaces are conflicting, it suffices to disable one of them, not both).

A more elegant class of solutions to resolving a duplicate address conflict would be for the node(s) which detect a conflict to ”negotiate” which interface should yield – possibly based on metrics such as active traffic flows for a given interface etc. This negotiation would take form of a broadcast of information (a ”CONFLICT” message), containing necessary information for a recipient to decide if it should yield and disable a given interface, or not.

V. Conclusion and future work

In this paper we have given an overview of the issues with duplicate address detection in OLSR networks. Using the proactive nature of the routing protocol, monitoring the routing control messages allows detection of many cases of duplicate addresses in the OLSR networks. This simple passive approach has, however, some shortcomings, both in terms of false alarms and in terms of undetected duplications under some very specific network conditions with exactly-symmetric networks. Thus, while passive duplicate address detection in OLSR networks provides a simple and overhead-free mechanism for detecting a large class of duplicate address conflicts, it might be beneficial to complement passive duplicate address detection with an active mechanism – possibly combined with an active mechanism for duplicate address conflict resolution.

References

[1] K. Mase C. Adjih. No overhead autoconfiguration olsr. Internet Engineering Task Force (IETF), Internet-Draft (work in progress), May 2005.

[2] T. Clausen and E. Baccelli. A simple address autoconfiguration mechanism for olsr. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005.

[3] T. Clausen and P. Jacquet. Rfc 3626: The optimized link-state routing protocol. Internet Engineering Task Force (IETF) Request For Comments, October 2003.

[4] R. Droms. Rfc 2131: Dynamic host configuration protocol. Internet Engineering Task Force (IETF) Request For Comments, March 1997.

[5] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical report, Project HiPERCOM, INRIA Rocquencourt, 2000. INRIA research report RR-3898.

[6] S. Thomson and T. Narten. Rfc 2462: Ipv6 stateless address autoconfiguration. Internet Engineering Task Force (IETF) Request For Comments, December 1998.