

Passive Duplicate Address Detection in Mobile Ad Hoc Networks

Kilian Weniger
Institute of Telematics
University of Karlsruhe
76128 Karlsruhe, Germany
Email: weniger@tm.uka.de

Abstract— Auto-configuration of nodes is an important issue in self-organizing mobile ad hoc networks (MANETs). Especially the property of unique addresses is critical for the main task of a data network: The delivery of packets to the intended destination. Stateless auto-configuration allows a node to construct an address on its own. Duplicate Address Detection (DAD), usually done by sending a query to the chosen address and waiting for a response, can be used to ensure the uniqueness of this address. The other approach, the distributed assignment of a priori unique addresses, can also be a bandwidth consuming task in a dynamic environment. In both cases, a merger of two configured networks is very difficult to detect and can lead to duplicate addresses. Thus, a continuous and bandwidth-efficient duplicate address detection mechanism would be eligible. In this paper, the feasibility of a new DAD approach is investigated: The detection of duplicate addresses in a passive way, only by monitoring routing protocol traffic. Based on classic link state routing, three concepts of Passive Duplicate Address Detection (PDAD) are proposed. Two link-state protocols currently in discussion in the IETF MANET working group, the Fisheye State Routing (FSR) protocol and the Optimized Link State Routing protocol (OLSR), are analyzed regarding these concepts. Finally, first simulation results are presented.

I. INTRODUCTION

Mobile ad hoc networks are infrastructure-free, dynamic wireless networks. Every node has routing capabilities in order to be able to forward packets to distant nodes. Usually, these networks are non-administered and self-organizing and, subsequently, require address auto-configuration.

In general, address auto-configuration mechanisms can be classified in stateful and stateless approaches. Stateful auto-configuration mechanisms assign a priori unique addresses to the nodes by maintaining a common pool of addresses. A popular stateful mechanism known from the Internet is the Dynamic Host Configuration Protocol (DHCP) [1]. In contrast, stateless auto-configuration allows the nodes to construct addresses based on a locally available ID (e.g. the MAC address) or random number generation. These addresses are not unique a priori and require Duplicate Address Detection (DAD). Even auto-configured IPv6 addresses, which are usually based on an

hardware ID, are not necessarily unique. Various reasons exist, why duplicate addresses may occur:

- The IP address is chosen randomly, because an hardware ID is not available or the hardware ID is too big to be embeddable in the IP address
- Parts of the IP address is randomly generated due to privacy reasons [2]
- The network is heterogeneous in terms of the network devices and no globally unique hardware ID exist
- The network device offers the possibility to modify the hardware ID
- Non-IEEE network interface cards are used that do not have a registered MAC address
- A failure in the manufacturing process leads to duplicate MAC addresses [3]

Especially in mobile networks, where nodes meet hundreds or thousands of other nodes in their lifetime, the probability of a conflict is higher than in a static network like a LAN. And even there, the IETF recommends to perform DAD on all auto-configured IPv6 addresses [4].

Due to the node mobility in MANETs, configured networks can partition and merge. These events are difficult to detect and can lead to duplicate addresses for both, stateless and stateful approaches. Thus, a continuous and bandwidth-efficient DAD is desirable. The basic idea of PDAD is, to achieve this goal by continuously monitoring routing protocol traffic.

Routing protocols for MANETs can be classified in reactive and proactive approaches. While reactive protocols only discover and maintain routes that are needed for data delivery, proactive protocols continuously maintain routes to all nodes in the network. In this paper, the focus is on proactive link state routing protocols.

The rest of the paper organized as follows: Section II summarizes related work. Section III classifies various mechanisms to detect duplicate addresses. Passive Duplicate Address Detection (PDAD) is introduced in section IV and three approaches are presented for classic link state routing. In section V, two routing protocols currently in discussion in the IETF are analyzed regarding PDAD. Section VI presents first simulation results. Finally, section VII concludes the paper.

II. RELATED WORK

Address auto-configuration can be classified in stateful and stateless auto-configuration. Stateful auto-configuration mechanisms assign a priori unique addresses to the nodes by maintaining a common address pool. DHCP [1] known from the Internet uses a central entity to maintain this pool and, thus, cannot be used in a peer-to-peer style mobile ad hoc network. The MANETconf [5] protocol tries to adapt this scheme to mobile ad hoc networks by using a mutual exclusion algorithm to maintain a distributed pool of addresses. Depending on the mobility scenario, the maintenance of a common address pool at all nodes in the network may be a complex and bandwidth consuming process, especially in the presence of frequent network partitioning and merging. Boleng [6] uses a similar approach in conjunction with a variable-length network layer addressing scheme.

Instead of the assignment of addresses by a second entity, stateless auto-configuration allows the nodes to construct addresses by themselves, usually based on a hardware ID or a random number. A DAD mechanism is used to assure the uniqueness of the address. The IETF zeroconf working group is working on such a mechanism for IPv4 [7] and the IP Version 6 working group already standardized a stateless auto-configuration mechanism for IPv6 [4]. Both protocols were not designed for mobile ad hoc networks, but adaptations exist [8] [9]. In [8], a node floods the network with an address request message addressed to the constructed address. If no reply is received before a timer expires, it is assumed that the address is not occupied. Because network merging is not considered, duplicate addresses can still occur. In [9], network merging is supported. But although an hierarchical approach is used, a considerable amount of bandwidth is needed solely for the detection of duplicate addresses.

Weak Duplicate Address Detection [10] aims at lowering the overhead needed for the DAD by integrating it with the routing protocol. Nodes in the network are identified not only by the IP address, but additionally by a key, which can be based on a hardware ID or a random number. This concept is similar to the scheme of embedding the MAC address in an IPv6 address, but with the difference that the key is not used for routing decisions. If a node receives a packet containing an IP address that is stored in its routing table, but with a different key, an address conflict is detected. Beside the fact that additional bandwidth is needed for the distribution of the keys, a conflict is not detectable if two nodes with the same address choose the same key, because the key is only generated once by each node.

III. CLASSIFICATION OF DUPLICATE ADDRESS DETECTION MECHANISMS

There are different DAD mechanisms that differ in when and how duplicate addresses are detected. To the best of our knowledge, all existing approaches for mobile ad hoc networks distribute additional information in the network [8] [9] [10]. This can be named Active Duplicate Address Detection

(ADAD). In contrast, PDAD tries to detect duplicates without disseminating additional control information.

In case a conflict is detected, at least one node has to give up its IP address. Imagine a situation, where node A has claimed an address first and node B starts using the same address in the same network later on, e.g. after a network merger. Because the address was unique at the time node A started to use it, all packets destined for node A should be routed to node A. If the detection mechanism is unable to guarantee that, the mechanism can be named loose DAD. The period of time, during which packets may reach the wrong destination is called the period of vulnerability [10]. In contrast, strict DAD has a period of vulnerability equal to zero. Certainly, the optimum is a strict detection mechanism, but it may be worthwhile to tolerate short periods of vulnerability, if a lot of bandwidth can be saved. The consequences of short periods of vulnerability for applications running on the respective nodes still have to be investigated.

IV. PASSIVE DUPLICATE ADDRESS DETECTION (PDAD) FOR LINK STATE ROUTING

Three approaches are presented in the following sections, all are based on the properties of link state routing protocols. Nodes using classic link state routing inform other nodes about their neighborhood by periodically sending link state packets. Usually, packets contain sequence numbers to distinguish fresh from old routing information. Given these information, a node can calculate the shortest path to all nodes in the network using Dijkstra's shortest path algorithm.

A. Passive Duplicate Address Detection based on sequence numbers (PDAD-SN)

Most link state routing protocols use sequence numbers to distinguish fresh from old routing information. The idea of PDAD-SN is to exploit this property. It can be observed that nodes in a properly configured network obey the following rules:

- A node uses increasing sequence numbers
- A node uses each sequence number only once
- Two nodes do not have the same neighborhood at the same time, if they are more than two hops apart from each other.

Following these properties, two theorems can be stated that apply if no duplicate addresses exist. If one of these does not apply, an address conflict is present in the network.

- 1) Two messages with the same sequence number and source address are copies of the same message.
- 2) A node does not receive a link state packet with its own address as source address and a sequence number, which is higher than its own counter value. The only exception from this is a sequence number wrap-around. This situation handled in section IV-B.

According to theorem two, a node can detect a conflict if it receives a message with its own address as source address and a sequence number that is higher than its own counter

value. Some routing protocols allow nodes only to forward routing information with higher sequence numbers than recently received routing information from this address. In this case, node A having the same address as node B, but a higher sequence number cannot detect the conflict: The messages sent by node B do not reach node A. Only the node with the lower sequence number (or one of its neighbors) is able to detect the conflict. If node B now gives up its address, the conflict is resolved. Actually, this is a fair conflict resolution: If every node initializes its sequence numbers to zero if it starts using an address, the node that uses an address the longest time always keeps it. Unfortunately, a node may have intended to send a packet to node B. In this case, the packet is delivered to node A, because all nodes have updated their routing table entries due to the higher sequence number. Subsequently, the period of vulnerability is not zero. To prevent this, intermediate nodes would have to detect the conflict. If the sequence numbers of two consecutively received updates with the same source address are very different, the node could ignore the routing information contained in the update and trigger the conflict resolution. Beside the additional control traffic needed for the conflict resolution, it may be very difficult to define “very different” here. Especially in the presence of high packet loss or network partitioning two consecutively received packets from the same node may have quite different sequence numbers.

If the sequence numbers of node A and B are almost the same, the conflict cannot be detected based on theorem two. Instead, the first theorem can be exploited: Assuming a network with only two nodes A and B having the same address, a third node can detect the conflict, if it receives two link state packets with the same source address and the same sequence number, but different link states. But this only works, if node A and B do not have the same neighborhood, which means they must be more than two hops apart. Duplicate addresses in the two hop neighborhood must be detected by other means (see section IV-E). If more duplicate addresses are present in the network, node A and B can have the same neighborhood in terms of addresses, although the distance is more than two hops (see figure 1(a)). Nevertheless, all conflicts can be resolved, if at least one address is unique and it is assured that no duplicate addresses exist in the two hop neighborhood of each node.

Theorem 1: Assuming that all addresses are unique within a node’s two hop neighborhood and that at least one node in the network has a unique address, all address conflicts can be resolved due to the fact that at least two nodes with the same address have a different neighborhood.

Proof: If a network with only duplicate addresses is assumed with one node (node D) having a unique address, all of node D’s neighbors can resolve their conflicts, because at least one address (node D’s address) is unique in the neighborhood of these nodes. In contrast, if two nodes with the same address would have node D’s address in their neighborhood, they must be two hop neighbors. This contradicts the assumption. ■

Once the neighbors of node D have unique addresses, the conflict of their neighbors can be detected, etc. An example is shown in figure 1. All nodes have duplicate addresses, but unique addresses within their two hop neighborhood (see figure 1(a)). Nevertheless, because they all have the same neighborhood, the conflict cannot be detected according to theorem one. Once node D with a unique address is added (see figure 1(b)), the conflict of the nodes with address C can be detected and resolved, because both nodes now have a different neighborhood (see figure 1(c)). Finally, the nodes with addresses A and B can resolve their conflicts in the same way (see figure 1(d)).

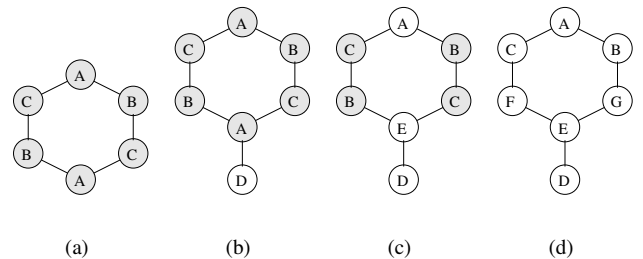


Fig. 1. Conflict resolution based on different neighborhood of nodes

B. Dealing with sequence number wrap-arounds

Sequence numbers are represented by a limited number of bits. Thus, a wrap-around must occur at some point of time. In this case the sequence number jumps from the highest possible value to zero. If no actions are taken, nodes don’t update their link states anymore, because they consider all received link state packets as outdated. To prevent this, very low sequence numbers can be interpreted as “greater than” very high sequence numbers.

In conjunction with the PDAD-SN, a sequence number wrap-around may result in the erroneous detection of duplicate addresses. Imagine node D sending a link state packet with the sequence number $S_1 = 65534$ to its neighbor G. Node G receives the message and forwards it to its neighbor, etc. After a wrap-around occurred, node D issues the next link state packet with the sequence number $S_2 = 0$. If node D now starts moving to the neighborhood of node F, it receives its own link state packet with the sequence number $S_1 = 65534$ (see figure 2). This breaks with theorem two of section IV-A and node D has erroneously detected an address conflict.

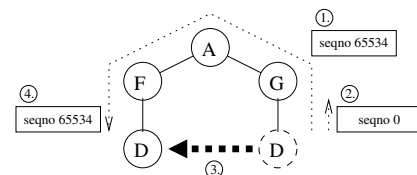


Fig. 2. Example of an erroneous conflict detection due to a sequencer number wrap-around

One way to prevent this situation is to suspend the PDAD-SN for nodes with sequence numbers close to the wrap-around region. This is expressed by equation 1.

$$S_2 < S_{thres} \wedge S_1 > S_{max} - S_{thres} \quad (1)$$

S_1 and S_2 are the sequence numbers of two consecutive update packets. S_{max} is the highest possible value of the sequence number. In case of a 16 bit sequence number, S_{max} is 65534. S_{thres} is the threshold and must be chosen high enough that an erroneous conflict detection cannot occur. The optimum value depends on the maximum time a message can travel in the network.

Due to the suspension of the detection mechanism, the period of vulnerability of nodes close to the wrap-around region increases with the value of S_{thres} . But if nodes initialize their sequence numbers to zero each time they start using an address, a sequence number wrap-around takes a long time, e.g., 2.7 years in case of FSR [11] with 24 bit sequence numbers and a five seconds update interval.

C. Passive Duplicate Address Detection based on the locality principle (PDAD-LP)

In case of link state routing protocols, the fact that nodes move with limited speed can be exploited. Usually, the frequency of routing updates is adjusted according to the maximum speed of the nodes. If routing updates are sent too frequently, bandwidth is wasted. On the other hand, if they are sent too infrequently, the nodes' topology informations are outdated and packets potentially won't find their way to the destination. Subsequently, the update frequency is usually chosen too high and most consecutive routing packets contain redundant information: The link states. In contrast, two nodes with the same address that are more than two hops apart have no link states in common, because they have a different neighborhood (see section IV-A and IV-E). Subsequently, if a node notices that most consecutive link state packets do not contain redundant link states, it is very likely that an address conflict exists.

There are situations where nodes receive consecutive packets with completely different link states although the source address is unique. This may happen, e.g., if a node joins the network, if it only has one neighbor and moves away from it. Thus, observing different link states once is not a sufficient requirement for the detection of an address conflict. Furthermore, in the presence of high packet loss or network partitioning, the likelihood of an erroneous conflict detection can increase dramatically. One solution is that different link states are ignored if the time between to consecutive updates is too long. A challenge is to discover this time and the optimal number of times different link states have to be detected before nodes conclude that a conflict exists in the network. These parameters can vary in respect to the routing protocol parameters and the mobility of the nodes. If the threshold is too high, duplicate addresses will not be detected. If it is too low, unique addresses may be erroneously interpreted as duplicates,

which is even worse. Thus, an adaptive threshold is desirable, e.g. by interpreting the behavior of the majority of nodes as normal and deviations as abnormal behavior. Unfortunately, this only works if the majority of nodes have unique addresses and can lead to high periods of vulnerability, which makes PDAD-LP inferior to PDAD-SN. On the other hand, PDAD-LP can be used even if no sequence numbers are available.

D. Passive Duplicate Address Detection based on the neighborhood (PDAD-NH)

Another possibility to detect duplicate addresses is to exploit the property that a node knows its own neighborhood and the neighborhood of the originator of a link state packet. If address A is unique, a node with address A only receives a link states containing address A, if the originator was a neighbor of node A at the time the packet was sent. If this is not the case, an address conflict is present. Subsequently, the nodes must maintain a cache containing the addresses of recent neighbors. But in case the sender of the link state packet is a common neighbor of the nodes with the same address, the conflict cannot be detected by PDAD-NH. Thus, conflicts in the two hop neighborhood must again be detected by other means (see section IV-E). A challenge is to choose the optimal timeout value for the cache entries. If the timeout is too high, memory is wasted and some conflicts may remain undetected. If it is too small, nodes erroneously detect conflicts. The value depends on the maximum time a message can travel in the network.

E. Providing unique addresses within the two hop neighborhood

As discussed in last sections, guaranteeing unique addresses in the two hop neighborhood is necessary to detect conflicts based on the link states. Furthermore, the correct delivery of packets to the last hop also requires unique addresses in the two hop neighborhood. If two nodes with a distance of equal or less than two hops have the same MAC layer and network layer address (e.g. a duplicate IPv6 address), they both receive and process a packet forwarded by a common neighbor and destined for the common address.

Unique addresses within the two hop neighborhood can be provided by extending the neighbor sensing function of the routing protocol. Most protocols use hello messages sent periodically by each node to maintain an up-to-date list of link states. By including a so-called Random Source ID (RSID), messages of two nodes with the same address can be distinguished. The RSID is a random number that changes for every hello message sent. Thus, two nodes do have different values at some point of time. RSIDs are also used in [9] and are similar to the MAC-keys in [10]. To detect duplicate addresses in the two hop neighborhood, the nodes also have to include the RSID of their neighbors in the hello message (see figure 3). If a node receives a message from its own address with an RSID that it did not use recently, an address conflict is detected.

Instead of explicit hello messages, some protocols use link state messages for neighbor sensing (e.g. FSR). In this case,

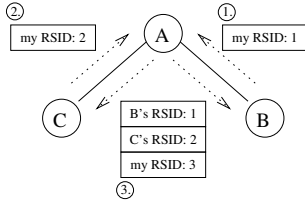


Fig. 3. Using Random Source IDs (RSIDs) to detect duplicate addresses in the two hop neighborhood

the mechanism must be adapted accordingly. In moderately dense networks, a small number of bits for the representation of the RSID, e.g. eight bits, is sufficient. Because the RSIDs are small and each node only sends the neighbors' RSIDs to its neighbors, the additional bandwidth consumed is low.

V. APPLICABILITY OF PDAD TO CURRENT IETF MANET ROUTING PROTOCOLS

In this section two link state routing protocols are discussed regarding the proposed concepts. Each protocol has its individual properties, which can affect the PDAD. Further study is needed to analyze the applicability to other routing protocols, especially reactive protocols.

A. Fisheye State Routing (FSR)

FSR [11] is a link state routing protocol designed with scalability in mind. Two innovative features are introduced: First, the fisheye technique has the effect that nodes send routing updates more frequently to closer destinations than to destinations far away. Second, the aggregation of link state packets saves bandwidth. Nodes collect all link state packets of their neighbors and send out only one aggregated packet per update period.

Unfortunately, the fisheye technique slows down the distribution of the link states of distant nodes. Subsequently, the period of vulnerability increases with the distance of the nodes having an address conflict to as much as 15 seconds per hop (default FSR parameters assumed).

Beside PDAD-NH, PDAD-SN can be applied, because the FSR uses sequence numbers. PDAD-LP should not be used, because the update interval may be too high for distant nodes, which can result in the erroneous detection of duplicate addresses.

B. Optimized Link State Routing (OLSR)

OLSR [12] uses the so-called Multi-point relay (MPR) concept to lower the routing protocol overhead. Therefore, only special nodes, the MPRs, issue and forward link state messages. In brief, the MPR selection algorithm works as follows: Nodes select a set of one hop neighbors as their MPRs, so that all two hop neighbors can be reached over these MPRs. A node that selects an MPR is called the MPR selector of that MPR. This concept also affects the PDAD. Assuming that node A and node B both have the same address, the following scenarios can occur:

1) Node A and node B are both MPRs

2) Only node A or node B is an MPR

3) Neither node A nor node B is an MPR

In case one, PDAD-SN and PDAD-LP can be used. Because the sequence number wrap-around mechanism of OLSR does not allow the proper detection of duplicate addresses, the mechanism described in section IV-B should be used instead. In case two, only one of the two nodes sends link state packets. This means that duplicate addresses can only be detected by PDAD-LP or by PDAD-SN if the MPR node has the higher sequence number. In case three, none of the nodes receive a link state packet with the source address of node A or B, thus neither PDAD-SN nor PDAD-LP can detect the conflict. In contrast, PDAD-NH can only duplicate addresses if they are included in link state packets. This is the case, if at least one node with a duplicate address is an MPR selector. Subsequently, by combining PDAD-SN, PDAD-LP and PDAD-NH, conflicts can be detected if at least one of the nodes with a duplicate address is an MPR or an MPR selector.

But because the MPR selection algorithm is affected by duplicate addresses, it may happen that routing updates are not properly propagated through the network, which makes the PDAD impossible. An example of such a scenario is shown in figure 4. Node A and E both have address 1 and node C and G both have address 3. Node C only knows of a two hop neighbor with address 1 and chooses node B as its MPR. Node E is in the same situation and chooses node F as its MPR. Subsequently, node D is not selected as MPR by any node and, thus, does not forward any link state packets. Node A, B and C never receive any link state packets of node E, F and G: The network is partitioned.

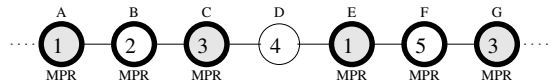


Fig. 4. Network partitioning due to MPR selection affected by duplicate addresses

To avoid this situation, all duplicate addresses within the four hop neighborhood of a node have to be detected. This can be done by extending the mechanisms presented in section IV-E to four hops, but with the side effect of big hello messages. On the other hand, if permanent node mobility is assumed, such scenarios resolve at some point of time and potential conflicts can be detected, but with an increasing period of vulnerability.

VI. SIMULATIONS

The applicability of PDAD-LP and PDAD-SN to FSR is investigated by simulating a moderately dense network with moderate mobility (see figure 5) using the GloMoSim network simulator [13]. Four cases can be distinguished: one/zero address conflict exists, while the nodes have the same/different sequence numbers. In case of a conflict both, the node with ID 40 and the one with ID 20 have the address 20 and in case of different sequence numbers node 40 has the higher sequence number. During the simulation, each node calculates

the similarity of the link states received from address 20 to the last known link state information of address 20. Old packets with lower sequence numbers are ignored.

| | |
|----------------------------------|--------------------------------|
| Number of nodes | 100 |
| Routing protocol | FSR |
| FSR scope/neighbor timeout | 2/15s |
| FSR intra-/inter update interval | 5s/15s |
| Simulation time | 600s |
| Area | 1500m \times 1500m |
| Mobility Model | Random Waypoint |
| RW parameters (min/max/pause) | $2\frac{m}{s}/2\frac{m}{s}/5s$ |

Fig. 5. Simulation parameters

Figure 6 shows the graph of node 20 monitoring the similarity of the link states received from address 20 to its own link states. In scenarios with unique addresses, the similarity of almost all packets is more than 80%. Only in case of duplicate addresses with different sequence numbers, entirely different link states can be detected frequently. This can be interpreted as an address conflict.

Figure 7 shows, how often the nodes detect different link states in updates from address 20. In accordance to the figure 6, node 20 detects entirely different link states very frequently in case the sequence numbers are different and address 20 is duplicate. In case the sequence numbers are equal, most nodes detect entirely different link states more frequently than in case of unique addresses. The difference is moderate, because the nodes only receive packets from both nodes with the same address and sequence number, if they are currently at the border where the link state updates with the same sequence number meet. Because the nodes and, subsequently, the border moves, they only detect entirely different link states during short periods of time in respect to the overall simulation time. In case of unique addresses, the nodes detect entirely different link states only once: at the time node 20 joins the network.

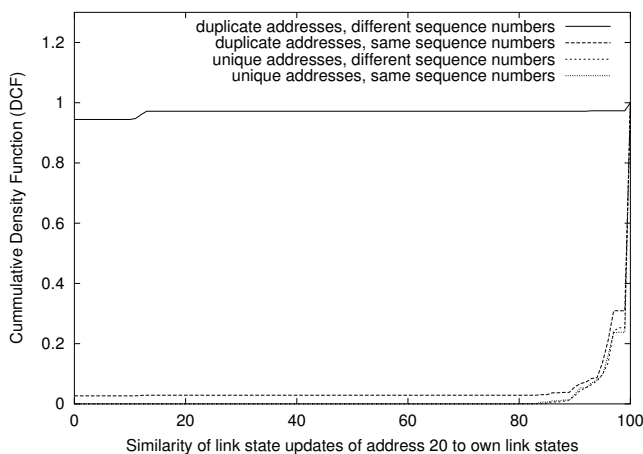


Fig. 6. Similarity of link state updates of address 20 to own link states

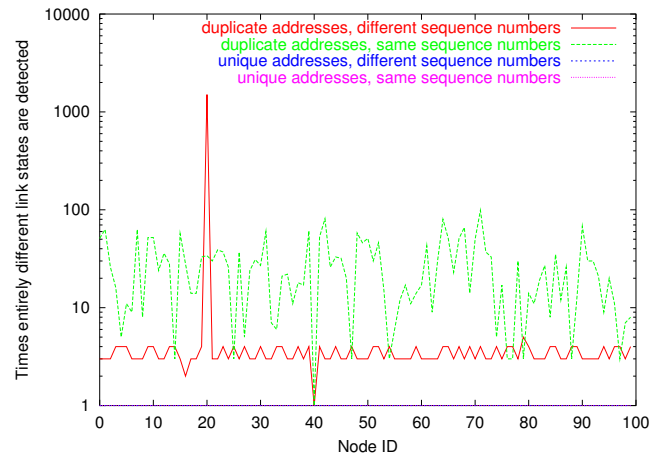


Fig. 7. Times entirely different link states are detected in consecutive link state updates of address 20

VII. CONCLUSION

This paper investigates the idea of detecting duplicate addresses in a passive way, only by monitoring routing protocol traffic. Based on classic link state routing, three concepts are presented. Following, the applicability of these concepts to OLSR and FSR is discussed. First simulations emphasize that Passive Duplicate Address Detection (PDAD) is feasible and that it can be applied to FSR. If certain periods of vulnerability can be tolerated, PDAD allows a continuous and bandwidth-efficient detection of duplicate addresses in a link state routed mobile ad hoc network.

ACKNOWLEDGMENT

This work was supported by the German Federal Ministry of Education and Research (BMBF) as part of the project IPonAir belonging to the research focus hyperNET. HyperNET stands for Universal Utilization of Communications Networks for Future Generations of Mobile Communications Systems.

REFERENCES

- [1] R. Droms, "Dynamic host configuration protocol," RFC 2131, Mar. 1997.
- [2] T. Narten and R. Draves, "Privacy extensions for stateless address autoconfiguration in IPv6," RFC 3041, Jan. 2001.
- [3] Intel. (1997) Duplicate mac address on intel stl2 server board. [Online]. Available: <http://www.intel.com/support/motherboards/server/stl2/ta-503.htm>
- [4] S. Thomson and T. Narten, "IPv6 stateless address autoconfiguration," RFC 2462, Dec. 1998.
- [5] S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc network," in *Proc. of IEEE Infocom 2002*, New York, USA, June 2002.
- [6] J. Boleng, "Efficient network layer addressing for mobile ad hoc networks," in *Proc. of International Conference on Wireless Networks (ICWN'02)*, Las Vegas, USA, June 2002, pp. 271–277.
- [7] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic configuration of IPv4 link-local addresses," IETF Draft, 2002.
- [8] C. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, "IP address autoconfiguration for ad hoc networks," IETF Draft, 2001.
- [9] K. Weniger and M. Zitterbart, "IPv6 autoconfiguration in large scale mobile ad-hoc networks," in *Proc. of European Wireless 2002*, vol. 1, Florence, Italy, Feb. 2002, pp. 142–148.

- [10] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in *Proc. of ACM MobiHoc 2002*, Lausanne, Switzerland, June 2002, pp. 206–216.
- [11] M. Gerla, X. Hong, and G. Pei, "Fisheye state routing protocol (FSR) for ad hoc networks," IETF Draft, 2002.
- [12] T. Clausen, P. Jaquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," IETF Draft, 2002.
- [13] (2001, Feb.) Glomosim: Global mobile information systems simulation library. [Online]. Available: <http://pcl.cs.ucla.edu/projects/glomosim/>