JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 821-833 (2018)
DOI: 10.6688/JISE.201807_34(4).0002

Enhanced Passive Duplicate Address Detection (EPDAD) for Autoconfiguration in MANETs

RESHMI TR[1] AND MURUGAN K[2]
[1]*School of Computing Science and Engineering*
VIT University
Chennai, 600127 India
[2]*Ramanujan Computing Centre*
Anna University
Chennai, 600025 India
E-mail: reshmi.engg@gmail.com; murugan@annauniv.edu

The uniqueness of IP addresses assigned to member nodes in MANETs is confirmed with a Duplicate Address Detection (DAD) method. But DAD often faces flooding issues which is addressed with a routing integrated protocol also called as Passive Duplicate Address Detection (PDAD) schemes. Although the existing scheme proves to be beneficial in terms of route processing time and overhead in the nodes they induce high complexity during integration. The paper presents a routing integrated and proficient approach for address autoconfiguration called the Enhanced Passive Duplicate Address Detection (EPDAD). The EPDAD employs a light weight and hastier technique for DAD. The EPDAD scheme uses a Bloom filter for compact storage of stateful information and easy retrieval during DAD. So as soon as the duplicates are detected by Bloom filters, it provokes routing integrated DAD procedures. The EPDAD and an existing scheme are simulated to study the performance of the schemes. The results prove that the EPDAD scheme has minimum DAD time, route processing delay and packet losses compared to the existing scheme.

Keywords: autoconfiguration, DAD, MANETs, IP address, routing integration, bloom filters

1. INTRODUCTION

The Mobile Ad-hoc Networks (MANETs) are wireless networks that self-configure and operate without any fixed infrastructure. Unique address configuration or assignment is a key challenge in MANETs due to the lack of infrastructure. The process or task called "autoconfiguration" facilitates the nodes to configure unique addresses and participate in the communication between the nodes. The autoconfiguration schemes require a distributed mechanism to ensure the characteristics such as sharing of unique addresses, management of stateful information, scalability, validation of rivals for same address *etc.* The flooding method called Duplicate Address Detection (DAD) is used to distribute collision-free addresses to the nodes in the network. The nodes initially configure temporary IP addresses in their interfaces and perform the DAD to confirm the uniqueness of the tentatively selected addresses. Once the address is proven to be unique, the tentative address is configured as the permanent IP addresses in the node interface. During network merging or if a new node enters the network, they are initially not allowed to route packets, but can communicate with the existing nodes to carry out DAD. Once the

Received July 4, 2017; revised August 16 & September 13, 2017; accepted December 15, 2017.
Communicated by Changqiao Xu.

node confirms to have a unique address, then they can send or receive data packets. So node passes through two states during autoconfiguration called pre-service and in-service. The node remains in pre-service until a DAD to ensure the uniqueness of its permanent address is completed. After acquiring the permanent IP address, the node enters into service and communicates with the other member nodes in the network. During network merging, it is very difficult to detect the duplicate addresses using pre-service DAD. Moreover, DAD deteriorates the performance of the network, causing high address configuration delay, control overhead, packet losses, *etc.* To avoid flooding routing integrated protocols or Passive Duplicate Address Detection (PDAD) schemes was introduced. But these schemes caused increase in the route processing time and overhead in nodes, thereby inducing complexity during integration. So even though the duplication detection of permanent addresses is mandatory, it is not checked by most of the schemes during routing.

The existing Passive Duplicate Address Detection (PDAD) scheme [1] uses live extraction and processing of routing information from routing tables which causes increased route processing delay, packet timeouts and forwarding delay. Even though these schemes avoid flooding, the control packets induce high complexity during integration. And also, these schemes consume more storage space for routing information and use a linear search algorithm for duplicate address detection. Moreover the existing schemes are either applicable to either proactive or reactive routing protocol. This paper proposes a routing integrated autoconfiguration scheme which deals with the existing challenges faced in the PDAD schemes. A routing integrated and proficient approach for address autoconfiguration called the Enhanced Passive Duplicate Address Detection (EPDAD) scheme is proposed. The EPDAD employs a light weight and faster technique for in-service DAD. It also uses a Bloom filter for compact storage of stateful information and easy retrieval during DAD. A Bloom filter is proven to be a straight forward space proficient data structure for representing a set, in order to resolve membership queries. A Bloom filter representation dramatically reduces storage space when compared to other data structures. In networks, the Bloom filters are proven to speed up or simplify packet analyzing and routing [2]. So if Bloom filters are used for storing a set of n addresses, it uses only an array of m bits for the representation rather than 2^m bits used in other data structures. To check whether an element, x belongs to the set, is just verifying the presence of the m bits corresponding to the Bloom filter [3]. So this also improves the speed for duplicate address detection.

The paper is organized as follows. The section 2 presents the literature survey of the autoconfiguration schemes. The section 3 discusses the proposed work and section 4 describes the performance analysis of the schemes. And finally the section 5 discusses the conclusions.

2. RELATED WORKS

Most of the schemes addressing the efficiency issues in detecting duplicate addresses paralyze the other performance of the schemes. The schemes assuring uniqueness of distributed addresses and their operations are discussed below. PACMAN [1] proposed by Weinger is a hybrid autoconfiguration scheme that allows nodes to self-generate IP addresses and perform DAD method for detecting duplicates. The scheme uses a

routing integrated method and handles duplicate detection during network merging and partitioning. The scheme uses a cross layer approach and maintains an allocation table to distribute addresses with less conflict probability. The scheme is dependent on the routing protocols and induces complexity due routing integration. Natalia *et al.* [3] presented a Filter-based addressing protocol (FAP) for autoconfiguration in MANETs. The scheme using sequence and Bloom filters was light weight and robust to packet losses. Further, the probabilistic analyses and the filter functionalities reduced address collision in autoconfiguration. The scheme proved to perform well in static and mobile network and manages network merging and partitioning. The scheme used a flooding technique for DAD and probably deteriorated the network performance.

Perkin *et al.* [4] proposed an address pool based stateful autoconfiguration scheme for MANETs. The protocol had two sets of address pools called temporary and permanent address pool. The new node configured with temporary address pool and was allowed to randomly choose another address from the permanent address pool. The chosen address was further checked for duplication by a flooding technique and it created high control overhead. Prophet addresses allocation [5] is a method based on a pseudo-random function with high entropy. The prophet is the first node in the network and it selects a seed for a random sequence and allocates addresses to the newly entering node. The prophet does not flood but the complex set of operations in pseudo-random number generator, to avoid duplicate addresses increased the protocol complexity.

Vaidya *et al.* [6] proposed weak duplicate address detection (WDAD) technique to avoid the flooding during duplicate address detection. The scheme embedded a unique key to each address and the routing packets were continuously monitored for duplicate address. The nodes declared as duplicates were not delivered packets. The scheme guarantee uniqueness of address, but the key embedded in the packets induced overhead to routing. Nesargi & Prakash [7] proposed an autoconfiguration scheme called "MANETConf". The scheme ensured distribution of unique addresses to the new nodes. The joining nodes named "requestor" requested reliable "initiator" nodes for acquiring unique addresses. The scheme handled network merging and partitioning by periodically broadcasting a verifying key. The scheme hence induced high overhead for its operation.

A stateless address autoconfiguration protocol (SLAAC) scheme was proposed for IPv6 networks by Thomas *et al.* [8]. The scheme assumed MAC identifiers are unique and hence the derived IP addresses using the same were also considered unique. But the flooding technique for duplicate detection made the scheme non-scalable. A hierarchical SLAAC scheme was proposed by Weniger & Zitterbart [9]. The scheme divided the network into many logical groups, and a leader node was selected in each subnet. The leader node selection at regular intervals or during topological changes induced high overhead in the network.

Wang & Qian [10] introduced a tree based autoconfiguration scheme. The hierarchical scheme assured address uniqueness and support network merging and partitioning. The scheme avoided message flooding by limiting the exchanges to 1-hop. But the nodes identified as isolated, indulged total reconfiguration until all nodes in its tree were configured and hence disrupted the total functioning of the scheme. Zhong & Siva [11] proposed an approach for IPv6 address autoconfiguration in ad hoc networks. The SLAAC and Neighbor Discovery Protocol (NDP) were extended for the usage in ad hoc networks. The protocol was simple and efficient, but required flooding of messages in the network

that deteriorates network with high overhead.

Sonia *et al.* [12] proposed a dynamically distributed address autoconfiguration protocol (DAACP) using the disjoint address space. The scheme used a fault tolerant address allocation approach to ensure the uniqueness of addresses by handling uncertainties such as node failures, network partitioning and merging. The scheme configured the newly entering nodes with the help of initiator nodes. The storage space required to store the address information (stateful information) for distributing unique addresses, managing network merging and partitioning, *etc.* causes high consumption of resources.

A scalable address autoconfiguration protocol called SAAMAN was proposed by Syed *et al.* [13]. The protocol automatically configured a network by allocating unique IP addresses to the nodes with low overhead and minimal processing cost. The scheme allocated Duplicate IP address Detection Servers (DDS) in the network, and checked the uniqueness of IP address. The DDS maintained IP addresses and stateful information to check the duplicates, so these servers consumed huge memory space for storing the information. Villalba *et al.* [14, 15] introduced two schemes called Distributed Dynamic Host Configuration Protocol (D2HCP) and Enhanced Distributed Dynamic Host Configuration Protocol (ED2HCP). The autoconfiguration protocols exploited the Optimized Link State Routing Protocol (OLSR) for synchronization. It ensured the uniqueness of the IP addresses under various network conditions such as link failure, message losses and network partition. The schemes were applicable only for link state protocols. As stated in survey [21], many of the similar autoconfiguration proposals [16-20] are proven to be complex and tedious.

3. PROPOSED WORK

The basic idea of EPDAD exploits the routing protocol events that either: (i) initiate an event in case of duplicate addresses, but not in the case of unique address and (ii) initiate an event in case of duplicate addresses, but rarely in the case of unique address. The stateful information in the scheme is compactly stored in Bloom filters which saves storage memory. Moreover the Bloom filters use a hastier DAD check, as the mapping is quick with the packet headers. Hence usage of bloom filters reduces route processing delay and packet retransmissions caused by packet timeouts. The EPDAD scheme also ensures uniqueness of addresses during network merging by deeply analyzing the packets.

3.1 Architecture

The architecture of EPDAD scheme shown in Fig. 1 is integrated with (Ad hoc On-Demand Distance Vector (AODV) and OLSR routing protocols and is implemented in network layer. The design elements and operations of EPDAD scheme are described in subsequent sections.

3.1.1 Routing protocol parser

The Routing Protocol Parser works at the network layer of TCP/IP stack. It is integrated with AODV and OLSR routing protocol modules to extract the routing events and updates. The routing events and node repositories (shown in Fig. 2) of AODV and OLSR

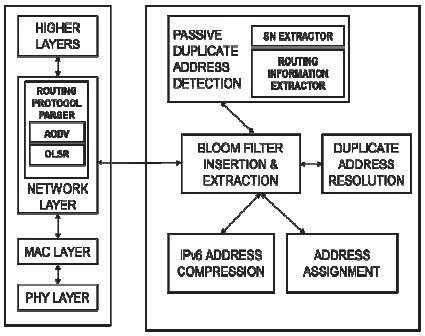
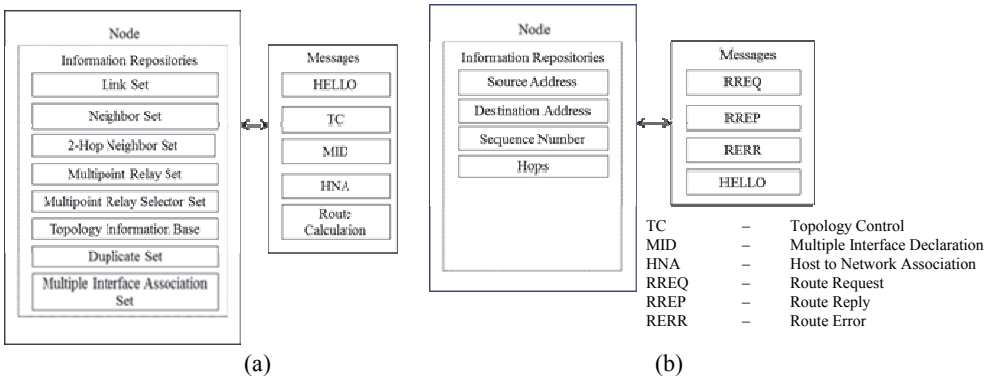


Fig. 1. Architecture of EPDAD scheme.



(a) (b)

Fig. 2. Node repository and routing packets in (a) OLSR (b) AODV protocols.

are analyzed based on their operations. The extracted information is sent to Bloom Filter Insertion and Extraction module for compact storage and DAD.

3.1.2 Bloom filter insertion and extraction

The stateful information extracted from Routing Protocol Parser is stored in Bloom filters. The Bloom filter is composed of an L bit vector bitmap (varies based on network size) which represents the set of available addresses. The Bloom filter inserts the extracted addresses that are compressed by IPv6 Address Compression module. Then the address is inserted into the Bloom filter using eight independent HASH function such as $h1$, $h2$ ... $h8$ and the corresponding output bits of the L bit vector (L value can be varied to ensure scalability) are set to 1. The existence of the address is verified in the Bloom filter by Duplicate Address Resolution module. To verify if an address already exists, the corresponding bits in the vector are checked. During the search, if any one of the bit is set to 0, then the address is considered unique. Otherwise, the search confirms the address being duplicated. The detected duplicate is reconfirmed using PDAD.

3.1.3 Duplicate address resolution

The Bloom filter is queried for DAD by checking whether an IP address is already in use. The query on the same eight HASH functions calculated over the input and the

corresponding bits in the bitmap is used to confirm the duplicates. The hashing functions $h1$, $h2$, …, $h8$ are verified in the filter. The match of all the hash function confirms the existence of an address. If any of the matches fails, the address is confirmed as not existing and the proceeding searches are declined. This method of duplication detection reduces the search time and enhances the speed of routing processes. If an address is proved to exist, the Duplicate Address Resolution initiates further detection and resolution by initiating an event for provoking PDAD.

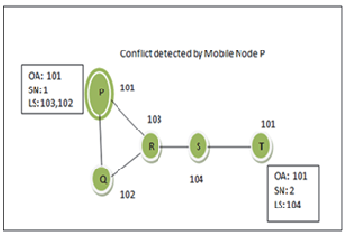
3.1.4 Passive Duplicate Address Detection (PDAD)

The Bloom filter significantly reduces space and time during DAD and routing, but the potential effects of false positives urge a reconfirmation of duplicates detected. The PDAD is prompt to reconfirm the duplicates and initiate resolution. This module includes SN and Routing Information Extractors which use few detection methods to confirm duplicates. The detection methods used in extractors are given below:

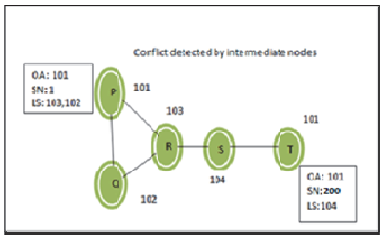
3.1.4.1 Sequence Number (SN) Extractor

The SN Extractor initiates detection methods based upon the sequence number embedded in the packets. The detection methods in the extractor are listed below:

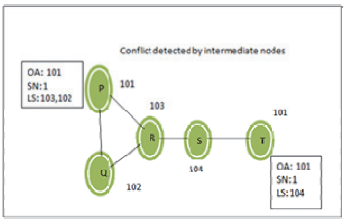
- Sequence Number (SN): Each routing packet sent by the node contains an SN that increment in each hop and SN method uses this SN to detect duplicate address in the network. In Fig. 3 (a) the scenario has two nodes, P and T with the same address 2001:db8::101. Consider node T receiving a routing protocol packet with an originator address (OA) equal to its own address (2001:db8::101) and a slightly lower SN than its own SN. So it cannot decide whether the packet is recently sent by itself or by another node with the same address. But if node P receives a packet with OA 2001:db8::101 and an SN 2, that is higher than its SN 1, then the packet is confirmed to be sent by another node with the same address.
- Sequence Number Difference (SND): The method uses SND for conflict detection based on difference in sequence numbers and is illustrated in the scenario given in Fig. 3 (b). The scenario has nodes, P and T sharing the same address, but having significantly different sequence numbers. The SND is higher than the maximum possible increment within the time when packets are received. Hence, these packets are not definitely sent by the same node and the intermediate nodes detect node's duplicate OA.
- Sequence Numbers Equal (SNE): The nodes may receive packets from different sources with the same addresses and sequence numbers. So the SNE method checks the link state information of these nodes to detect the duplicates. The method is illustrated in the scenario given in Fig. 3 (c). The scenario shows nodes, P and T with the same addresses and sequence numbers. If an intermediate node receives multiple packets with same OA and sequence numbers, then they are confirmed as packets received from different nodes of different link states. Therefore the link state information is analyzed by intermediate nodes to detect duplication of OA.
- Sequence Numbers Always Increment (SNI): SNs in packets are incremented for each packet. But if the node receives HELLO packets from the same address and with lower SN, the duplicates are detected by verifying the two-hop neighborhood information.



(a) Conflict detection using SN.



(b) Conflict detection using SND.



(c) Conflict detection using SNE.

Fig. 3. Methods in SN Extractor.

3.1.4.2 Routing information extractor

The Routing Information Extractor extracts the information from routing packets for duplication detection. The duplication detection methods in AODV and OLSR protocols are different based upon their operations and are discussed below.

- Source Address (SA): This method is applicable for both AODV and OLSR. It uses IP header information of routing packets to detect duplicates. Conflict in neighbor node addresses is detected when packet SA is equal to the destination address.
- Duplicate Cache (DC): The method is applicable for both OLSR and AODV and makes use of IP header information in routing packets to detect duplicates. Conflict is detected when two nodes have same SN, OA and SA.
- RREQ-Never-Sent (RnS): The RnS method is specially used for AODV protocol. If a node receives a RREQ from a similar address, which is never sent from its own address, then the node confirms the duplication of its address.
- RREP-Without-RREQ (RwR): The RwR method is used only in AODV protocols. If a node receives RREP without RREQ, then the node confirms duplication of its address.
- 2RREPs-on-RREQ (2RoR): The 2RREPs-on-RREQ (2RoR) method used in AODV detects a conflict, if a node receives more than one RREP for the same RREQ packet.

The PDAD employs SN and Routing Information Extractors for confirming duplicate addresses. The SA, DC, SN, SND, SNE, and SNI methods are used in OLSR and SA, DC, RwR, RnS, 2RoR, SN, SND, SNE, and SNI methods are used in AODV for confirming the duplicates. The duplicate addresses are resolved by sending Address Conflict Notification (ACN) message. The ACN message is sent to the recently joined node with duplicate address. The Address Assignment initiates the resolution of duplicate by assigning new addresses.

3.1.5 Address assignment

The duplicate confirmed by PDAD, provokes the Address Assignment module to generate conflict free address to the node. A random number generator is employed to generate random addresses to reduce the possible address duplications. The random addresses generation is based upon the key of the node, duplicated address and the attempt for regeneration. The probability of duplication as expressed in well-known birthday paradox [4] is proved to be significantly low. The PDAD informs the recently joined node to reconfigure its address using an ACN message and Address Assignment assigns the address generated using an Allocation message. The node receiving the Allocation message acknowledges the receipt with an Address Receipt and reconfigures its interface with the new address. Subsequently, an event is initiated in the Bloom filter to insert the new address and update the stateful information.

3.1.6 IPv6 address compression

The IPv6 Address Compression module compresses the IPv6 addresses generated and assigned by the Address Assignment and initiates an event in Bloom filter Insertion and Extraction. The IPv6 addresses of 128 bits are compressed to 24 bits by employing a simple compression technique [5]. Each three bits of the compressed bits are introduced independently to eight HASH functions. The HASH resultant returns respective addresses of bits which are set to '1' in the bitmap for address insertion. The architectural modules of EPDAD scheme such as Routing Protocol Parser, Bloom filter Insertion and Extraction, Duplicate Address Resolution, PDAD, Address Assignment and IPv6 Address Compression are interdependent, and provoke events based on scenarios. Fig. 4 shows the address insertion and extraction in EPDAD scheme.

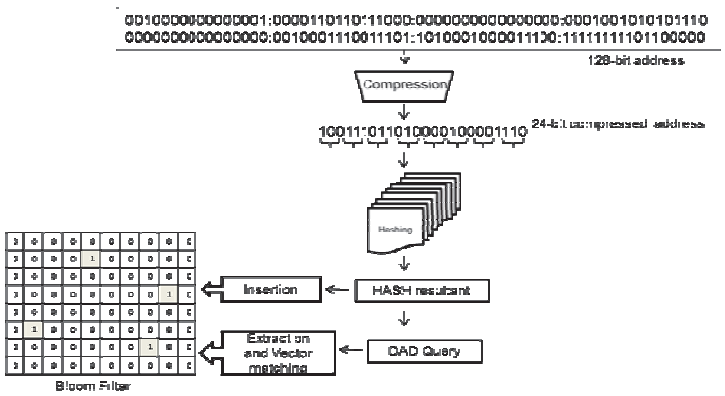


Fig. 4. Address insertion and extraction in the EPDAD scheme.

4. PERFORMANCE EVALUATION

The performance of EPDAD and the existing PDAD schemes [1] are studied and evaluated using Network Simulator (NS-2) [22]. The OLSR and AODV protocols in the

scheme were extended to support duplication during routing, address change in between routing, extraction of live routing traffic, *etc.* The message formats used for the simulation were structured as directed in the Message Format standard [23]. The parameters used for the implementations are given in Table 1. The simulations were carried out for the duration of 50 seconds and the results were taken by averaging over 20 runs with different seeds. The time interval between the messages was set to 0.15 seconds and the evaluations were done by varying the node population in the networks.

Table 1. Simulation parameters.

Simulation area	1250 m × 1250 m
Number of nodes	50-250(variable)
Mobility Model	Random way point
MAC Model	802.11
Transmission Range	250 m
Node mobility	10 m/s
Protocols	EPDAD-AODV, PDAD-AODV EPDAD-OLSR, PDAD-OLSR

The simulation focuses on the following four performance metrics for the analysis:

Protocol Overhead: The overhead is the total number of control and maintenance packets used by the schemes. The control packets incurred by EPDAD and PDAD schemes were compared and plotted as shown in Fig. 5. A linear search of the 128 bit addresses in routing tables causes delay in route processes of PDAD schemes. This results in more packet timeouts and drops as well as in extra overhead. The use of Bloom filter enhances the speed of DAD that facilitates quick packet processing and routing. Hence the packet drops due to timeouts are prevented and overall packet exchanges are reduced. The results showed that the packet exchanges in EPDAD-AODV scheme were 7.55% lower than those of PDAD-AODV scheme. Similarly Fig. 6 shows that EPDAD-OLSR outperforms PDAD-OLSR with a 2.34% less overhead. These reductions in overhead packets show the decrease in packet losses and retransmissions in EPDAD schemes.

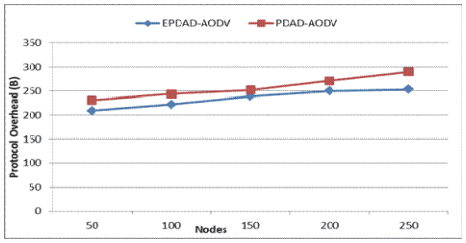


Fig. 5. Protocol overhead of EPDAD and PDAD in AODV as a function of node population.

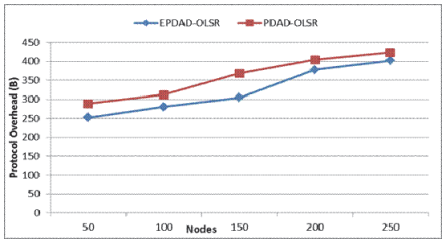


Fig. 6. Protocol overhead of EPDAD and P-DAD in OLSR as a function of node population.

Packet Delivery Time: It is defined as the average time taken by the packets to reach the destination addresses from the source addresses. In passive DAD schemes, the routing packets are monitored in each node for DAD and then forwarded for routing. So the

packets are detained at each node for a certain time interval for processing and forwarding. In PDAD scheme, the linear search of the IP address and DAD from extracted information in routing tables cause more delay compared with EPDAD. In EPDAD, the Bloom filter based DAD reduces the packet processing time by enhancing the search speed in each node. Hence the overall packet delivery time from source to destination is comparatively very low in EPDAD schemes. Fig. 7 shows that the packet delivery time in EPDAD-AODV protocol is 31.23% low compared with PDAD-AODV protocol. Similarly, Fig. 8 shows that EPDAD-OLSR reduces the packet delivery time to 15.89% compared with PDAD-OLSR. The performance of EPDAD scheme is exaggerated by replacing the linear search with Bloom filter search.

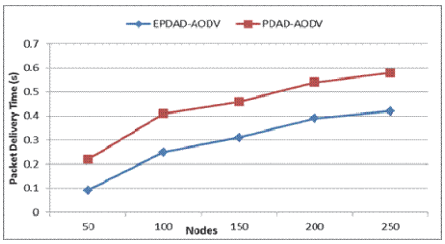


Fig. 7. Packet delivery time of EPDAD and PDAD in AODV as a function of node population.

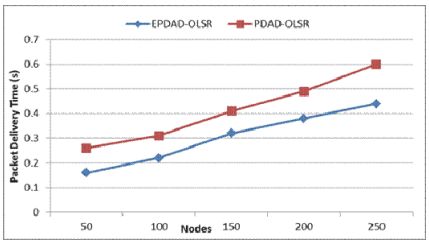


Fig. 8. Packet delivery time of EPDAD and PDAD in OLSR as a function of node population.

Duplication Detection Time: The duplicate detection time is the time taken by the scheme for detecting a duplicate IP address and initiating resolution. The in-service DAD schemes are relatively slow to detect duplicate addresses compared with pre-service DAD scheme because of its routing integrated detection methodology. The linear search and the routing integrated DAD in PDAD schemes cause extra delay in DAD. The EPDAD scheme uses the Bloom filters for the initial duplicate detection and invokes more detection algorithms for confirming duplicates. So the efficient use of Bloom filters for duplicate detection eases the DAD in EPDAD and hence a thorough analysis of each routing packet for duplicate detections as in PDAD scheme is avoided in EPDAD scheme. Figs. 9 and 10 showed that the duplicate detection time of EPDAD-AODV was 58.23% lower compared with PDAD-AODV and EPDAD-OLSR was 44.89% lower than PDAD-OLSR schemes.

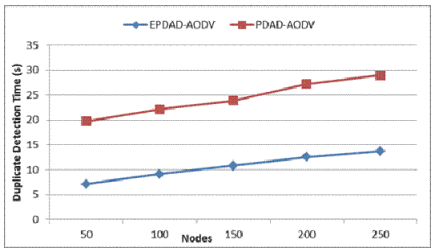


Fig. 9. Duplication detection time of EPDAD and PDAD in AODV as a function of node population.

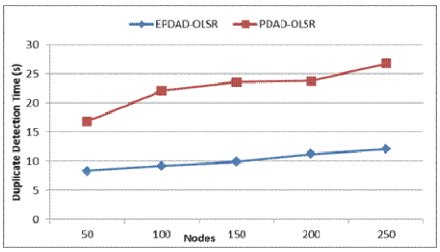


Fig. 10. Duplication detection time of EPDAD and PDAD in OLSR as a function of node population.

Packet Loss: The total number of control and maintenance packets lost during the working of the scheme is defined as packet loss. The routing packets in nodes are analyzed for duplicate detection in in-service DAD schemes. The packet processing time is an important factor influencing the packet delivery time. Hence the scheduled packet deliveries with minimum time avoid packet timeouts. In EPDAD schemes with compact insertion and easy extraction of stateful information to and from Bloom filters eases the DAD and minimize the time consumed for forwarding the packets as shown in Figs. 11 and 12. So the packet losses and retransmissions are mostly avoided in EPDAD schemes.

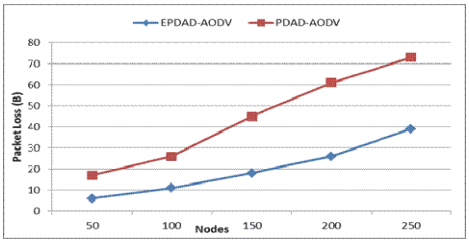


Fig. 11. Packet loss of EPDAD and PDAD schemes in AODV as a function of node population.

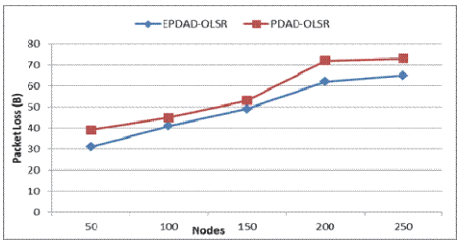


Fig. 12. Packet loss of EPDAD and PDAD schemes in OLSR as a function of node population.

5. CONCLUSIONS

The routing integrated or in-service DAD scheme called EPDAD scheme uses very less control packets, and thereby reducing protocol overhead during the working of the autoconfiguration scheme. The passive nature of address duplicate detection in EPDAD scheme is proved to perform well for in-service DAD. The duplicate address of nodes participating in routing after network merging, is also well handled by the scheme. The EPDAD extracts the address information from routing packets and node repositories and store it in the Bloom filter for conflict resolution. The stateful information stored in the Bloom filter facilitates efficient DAD and compressed address storage. The duplicate detection using Bloom filters greatly reduces the packet processing delay, and thereby routing packets quickly. Although Bloom filters allow false positives causing further proceeding to PDAD for duplicate confirmation, the space savings, fast route processing and duplicate detection prevail over this drawback. The enhanced filter based technique improves the overall routing efficiency.

REFERENCES

1. K. Weniger, "PACMAN: Passive autoconfiguration for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 23, 2005, pp. 507-519.
2. A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, Vol. 1, 2004, pp. 485-509.
3. C. F. Natalia, D. Marcelo, M. Donato, and C. M. Otto, "An efficient and robust addressing protocol for node autoconfiguration in ad hoc networks," *IEEE/ACM Tran-*

sactions on Networking, Vol. 21, 2013, pp. 845-856.
4. C. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, "IP address autoconfiguration for ad hoc networks," *IETF Internet Draft*, http://www.tools.ietf.org/html/draft-perkins-manet-autoconf-01, 2001.
5. H. Zhou, L. M. Ni, and M. W. Mutka, "Prophet address allocation for large scale MANETs," *Ad Hoc Networks*, Vol. 1, 2003, pp. 423-434.
6. N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2002, pp. 206-216.
7. S. Nesargi and R. Prakash, "MANETconf: Configuration of hosts in a mobile ad hoc network," in *Proceedings of the 21st Annual Joint Conference of IEEE Computer and Communications Societies*, Vol. 2, 2002, pp. 1059-1068.
8. S. Thomson, T. Narten, and T. Jinmei, "IPv6 stateless address autoconfiguration," RFC 4862, 2007.
9. K. Weniger and K. Zitterbart, "IPv6 autoconfiguration in large scale mobile ad-hoc networks," *European Wireless Proceedings*, Vol. 1, 2002, pp. 142-148.
10. X. Wang and H. Qian, "A tree-based address configuration for a MANET," *Pervasive and Mobile Computing*, Vol. 12, 2014, pp. 123-137.
11. Z. Fan and S. Subramani, "An address autoconfiguration protocol for IPv6 hosts in a mobile ad hoc network," *Computer Communications*, Vol. 28, 2005, pp. 339-350.
12. G. M. Sonia, E. Amine, and F. Kamoun, "Distributed address auto configuration protocol for Manet networks," *Telecommunication Systems*, Vol. 44, 2010, pp. 39-48.
13. H. R. Syed, S. Saha, and A. Rahman, "SAAMAN: scalable address autoconfiguration in mobile ad hoc networks," *Journal of Network and Systems Management*, Vol. 19, 2011, pp. 394-426.
14. L. G. Villalba, J. G. Matesanz, A. L. Sandoval Orozco, and J. D. Márquez Díaz, "Distributed dynamic host configuration protocol (D2HCP)," *Sensors*, Vol. 11, 2011, pp. 4438-4461.
15. L. G. Villalba, A. S. Orozco, J. G. Matesanz, and T. H. Kim, "E-D2HCP: Enhanced distributed dynamic host configuration protocol," *Computing*, Vol. 96, 2014, pp. 777-791.
16. M. Grajzer, T. Żernicki, and M. Głąbowski, "ND++ – an extended IPv6 neighbor discovery protocol for enhanced stateless address autoconfiguration in MANETs," *International Journal of Communication Systems*, Vol. 27, 2014, pp. 2269-2288.
17. C. Adjih, S. Boudjit, P. Jacquet, A. Laouiti, and P. Mühlethaler, "An advanced configuration and duplicate address detection mechanism for a multi-interface OLSR network," http://www-l2ti.univ-paris13.fr/~boudjit/Research/RR-INRIA/RR-5747.pdf, 2005.
18. X. Wang, H. Cheng, and Y. Yao, "Addressing-based routing optimization for 6LoWPAN WSN in vehicular scenario," *IEEE Sensors Journal*, Vol. 16, 2016, pp. 3939-3947.
19. T. R. Reshmi and K. Murugan, "Filter-based address autoconfiguration protocol (FAACP) for duplicate address detection and recovery in MANETs," *Computing*, Vol. 97, 2015, pp. 309-331.
20. T. R. Reshmi and K. Murugan, "Selective address allocator configuration protocol (SAACP) for resource constrained MANETs," *Journal of Information Science and*

Engineering, Vol. 32, 2016, pp. 197-211.
21. A. Munjal and Y. N. Singh, "Review of stateful address auto configuration protocols in MANETs," *Ad Hoc Networks*, Vol. 33, 2015, pp. 257-268.
22. Network Simulator: http:///www.isi.edu/nsnam/ns.
23. T. Clausen , C. Dearlove, J. Dean, and C. Adjih, "Generalized mobile ad hoc network (MANET) packet/message format," IETF RFC 5444, 2009.

Reshmi TR received her Ph.D. under the Faculty of Information and Communication Engineering at Anna University, Chennai, India. She is currently working as a Senior Assistant Professor in VIT University, Chennai, India. Her research area includes IPv6, MANETs, QoS, security and service applications.

Murugan K received his Ph.D. under the Faculty of Information and Communication Engineering at Anna University, Chennai. He is currently working as a Professor in Ramanujan Computing Centre, Anna University, Chennai, India. His area of interest includes mobile computing, MANETs and wireless sensor networks.