

Secure and Dynamic IP Address Configuration Scheme in MANET

Poulami Choudhury, Koushik Majumder and Debashis De

Abstract Mobile ad hoc network (MANET) is an infrastructure-less network having dynamic topology and volatile nodes connected by wireless communication. Secure communication in MANET demands unique IP address assignment to ensure proper routing of packets. It is challenging for a decentralized network to have dynamic and unique IP addressing scheme. In this paper, we focus on the difficulties of secure message passing and assigning unique IP addresses to new nodes willing to join in MANET. The message passing is authenticated using both symmetric and asymmetric keys, and arrival conflict is diminished by time stamp. Each node being a proxy server can allocate unique IP address to new node. Each node maintains a unique tuple of own IP address, node ID, and MANET ID for efficient network merging and partitioning. This scheme offers a secure and efficient mechanism of configuring a MANET.

Keywords Mobile ad hoc network • IP address • Routing • IP address configuration • Proxy server • Symmetric and asymmetric key

1 Introduction

MANET has very vibrant characteristics which demand unique identification of each node in network for source destination communication. There is a manual configuration scheme for configuring IP addresses in an ad hoc network. It is well

P. Choudhury (✉) · K. Majumder · D. De
Department of Computer Science and Engineering, West Bengal University of Technology,
Kolkata, India
e-mail: poulami.me.13@gmail.com

K. Majumder
e-mail: koushik@ieee.org

D. De
e-mail: dr.debashis.de@gmail.com

fitted in small-scale network but not for large network. Centralized server-based and fixed infrastructure network has a secure authenticated dynamic host configuration protocol (DHCP) server. As MANET is dynamic, the network might get partitioned at some times and might also merge. Therefore, manual configuration may lead to conflict of address. Centralized server DHCP cannot be used in a distributed network like MANET. For assigning an IP address, a standard IP addressing protocol should have the following objectives: dynamic IP address configuration, uniqueness, robustness, scalability, security, and adaptivity.

The rest of the paper is organized as follows: A brief note on the related research work in this area is given in Sect. 2. In Sect. 3, the proposed scheme is explained in detail. In Sect. 4, conclusions are presented.

2 Related Work Review

Address configuration schemes for MANET can be classified into three categories: neighbor-based schemes, decentralized schemes, and centralized schemes. The centralized scheme or leader-based allocations are DHCP, agent, or initiator-based allocation schemes. Most of the existing address allocation algorithms for a MANET use duplicate address detection (DAD) mechanism [1, 2] to resolve address conflict in the network. In neighbor-based schemes, a new node is configured by a neighbor node, so it does not suffer from network-wide flood or centralized control. In [3], a root node is also responsible for address reclamation and network merging. Therefore, if a root node fails, then the address configuration system may collapse. Ghosh et al. [4] proposed an ID-based address configuration scheme. The scheme assumes that in the initial state, the first new node in a MANET sets its address as 0. If more than one new node joins a MANET at the same time, then address conflict can occur.

In this paper, both the authentication and uniqueness are taken as a matter of concern for IP address allocation in distributed and dynamic MANET network.

3 Proposed Work

In this paper, we propose a secure and dynamic IP address configuration scheme. It is authenticated mutually, and uniqueness is guaranteed by using ID. We have given an improved solution to the problems that may arise due to host failures, message losses, mobility of the hosts, and network partitioning/merging.

Unique address allocation scheme: Our proposed address allocation algorithm has two parts: (a) for the new node (N_n) (*Algorithm 1*) and (b) for the proxy that assigns IP address (*Algorithm 2*).

A new node at first generates its public key using its hardware address and private keys using a random function and one-time session key. Then, it sends the

Discover message to 1 hop broadcast nodes. And a DiscTimer () is set on. If the DiscTimer expires and no existing nodes are present in MANET, then no Choice (CHOICE) message is received. In cases when such CHOICE message is not received or if the variable counter is greater than the threshold (value 2), then the new node calls self_configure (). This function sets the IP address of the node to 169.y.0.1 (for class B) and generates Node_id and MID using random function. On the other hand, if it receives multiple CHOICE messages with Priority (Pr_p) meant for generating IP address for child, then the highest priority proxy node is chosen by the new node. The priority is set depending on the available IP address in proxy recycle list and its father's recycle list. New node then sends request message REQ to that selected highest priority proxy node along with time stamp. This REQ message is encrypted using both symmetric key cryptography and asymmetric key cryptography. After receiving REQ message from proxy node, it will send a Reply (REP) message with unique IP address to the node N_n (Table 1).

Next, the N_n node generates its node ID using hash function on allocated IP address and public key and sets the MANET ID from father node. After that, the SELECT message is sent to the selected proxy node, and the REQTimer () is set off and acknowledgment timer (ackTimer ()) is set on (Table 2).

After receiving Select message from new node, the proxy n node sends an acknowledgment message along with the next IP value. And simultaneously, it broadcasts NextIP value to the nodes with same x.y.j.* (siblings). After getting the acknowledgment message from the selected proxy node, the new node is considered to be fully configured.

Algorithm 3 shows the steps for generating new unique IP address from IP address of Proxy, if available. Here, the range of IP address that the root proxy (169.y.0.1) can assign is from 169.y.1.1 to 169.y.254.1, and the range of IP address that other proxy can assign is from (i) 2 to 254 for each 169.y.j. as the NextIP (i) is broadcasted to the other nodes after each time of increase in NextIP (i). So there is no limitation for any proxy node to generate next IP. Proxy nodes have no limited range of IP addresses. This scheme requires synchronization between all the nodes in MANET. So the update of NextIP should be taken place consistently (Table 3).

Authentication scheme: Here, in combination of symmetric key cryptography and asymmetric key cryptography, the sender encrypts the message with the symmetric key (one-time symmetric key (K_s)) algorithm and transfers encrypted message and K_s . Sender also encrypts K_s with receiver's public key. This process is called key wrapping. Now, sender puts both the encrypted message and symmetric key together in digital envelop and sends the digital envelop to the receiver. Receiver receives, and after opening, the digital envelop uses the same asymmetric key algorithm as was used by sender and own private key to decrypt the logical box that contains the one-time symmetric key (K_s). Finally, receiver applies the same symmetric key algorithm as was used by sender and symmetric key (K_s) to decrypt the encrypted message. Each Request message has the priority, and the priorities of any two Request messages are compared according to the following algorithm: If the time stamp in one Request message T_n is earlier than the one in another Request message T_n' , then T_n has higher priority than T_n' .

Table 1 Algorithm 1: IP address allocation for new node N_n

| | |
|--|--|
| <pre> 1. begin 2. Set threshold $\leftarrow 2$; begin \leftarrow true; config \leftarrow false; counter $\leftarrow 1$; 3. Generate PublicKey $\leftarrow K_{n1}$; PrivateKey \leftarrow K_{n2}; SymKey $\leftarrow K_s$; // public key is the hard- ware address of the node 4. if begin == true and counter \leq threshold then 5. DISC = {my_Hw_add, K_{n1}} and Send to 1 hop broadcast; 6. Start DiscTimer(); 7. begin \leftarrow false; 8. else 9. self_configure(); //first node in MANET self- configure its IP address with X.Y.O.1(class B) 10. Config = true; 11. end 12. if more than one (SignP, CHOICE (Pr_p, PrIP, K_{p1})) then 13. Set all the Pr_p in a queue in descending order; 14. Select highest probability proxy from queue; 15. Generate SignN = (CHOICE(), K_{n2}); 16. if (SignP == SignN) then 17. Generate SignN(REQ, K_s); 18. Generate K_{wrap} (K_s, K_{p1}); //One time Key- wrap 19. Send (K_{wrap}, SignN, T_n) message to the proxy node with highest probability; // T_n is the time stamp of new node 20. Start ReqTimer(); 21. else 22. Select next highest probability proxy node from queue; 23. goto step 15; 24. end 25. if (REP+SignP+(T_n+1)) message from proxy node then </pre> | <pre> 26. Select OfferIP from REP message and check the TimeStamp ($T_n + 1$) nearest higher to its own TimeStamp; 27. Generate node_id_T = H(Pr_IP, K_{p1}); 28. Generate SignN(REP, K_{p1}); 29. if SignN == SignP and node_id_T == node_id_p then 30. Generate node_id_N = H(OfferIP, K_{n1}); // create node id using allocated IP address and public key 31. Set MID_n = MID_p; 32. Generate SignN (SELECT(node_id_N), K_{p1}); 33. Send (SELECT(node_id_N) + SignN) mes- sage to selected proxy; 34. Stop REQTimer(); 35. Start ackTimer(); 36. else 37. Select next highest probability proxy node from queue; 38. goto step 15; 39. end 40. if (ACK + SignP) is received from selected proxy then 41. Generate node_id_T = H(IP, KPP); 42. Generate SignG(K_{p1}, ACK); 43. if SigG == SignP and node_id_T == node_id_p then 44. stop ackTimer; 45. Config = true; 46. end 47. if timeout(DiscTimer) then 48. begin \leftarrow true; counter \leftarrow counter + 1; 49. if timeout(REQTimer) then 50. begin \leftarrow true; counter \leftarrow counter + 1; 51. if timeout(ackTimer) then 52. begin \leftarrow true; counter \leftarrow counter + 1; </pre> |
|--|--|

Departure of a node from MANET: A node in a MANET makes a request for graceful departure to its father node, and the father node allows the node to DEPART and update its recycle list by making the leaving IP address reusable. Dynamic topology, weaker wireless connection, and mobility of devices cause a node to be out of the radio range of MANET. Most of the time a node does graceless departure, which causes shortage of IP address. But by using HELLO message (message of AODV routing protocol) broadcast, a node gets the

Table 2 Algorithm 2: IP address allocation for existing proxy node

| | |
|--|--|
| 1. Begin 2. Set proxy_pub_key $\leftarrow K_{p1}$; proxy_pri_key $\leftarrow K_{p2}$; 3. if DISC() message is received from N_n then 4. Set Prob_IP_alloc $\leftarrow Pr_p$; 5. Generate SignP(CHOICE (Pr_p , PrIP, K_{p1}), K_{n1}); 6. Send (CHOICE, SignP) message to the new node; 7. end 8. if (K_{wrap} , SignN, T_n) message is received from node N_n then 9. Decrypt K_{wrap} using K_{p2} and get K_s ; <i>//asymmetric key cryptography decryption</i> 10. Generate SignP(REQ, K_s); 11. if SignP == SignN and T_n is closest time-stamp then 12. if free IP address is available in the RecycleList then 13. OfferIP = minimum IP from RecycleList; 14. Generate SignP (REP(OfferIP, con-fig_parameters), K_{n1}); 15. Send (REP+SignP+(T_n+1)) message to N_n ; 16. else if node is able to generate new IP then 17. OfferIP = generate_IP(my_IP); <i>// unique IP address generation from own IP address</i> 18. Generate SignP(REP(OfferIP, con-fig_parameters), K_{n1}); 19. Send (REP+SignP+(T_n+1)) message to N_n ; 20. else 21. Send the REQ_F message to Father node encrypted by pub_key of Father; 22. end | 23. if (REP_F(OfferIP)+SignF) message is received from father then 24. Generate node_id $_T$ = Hash(F_IP, K_{F1}); 25. Generate SignP using REP_F(OfferIP) and public key of Father; 26. if SignP == SignF and node_id $_T$ == node_id $_F$ then 27. Generate SignP(REP(OfferIP, con-fig_parameters), K_{n1}); 28. Send (REP+SignP+(T_n+1)) message to N_n ; 29. else 30. Generate SignP (REFUSE, K_{n1}); 31. Send (REFUSE + SignP) message to N_n ; 32. end 33. else 34. Drop the (REQ, SignN, T_n) message; 35. end 36. if (SELECT(node_id $_N$) + SignN) is received from N_n then 37. Generate SignG(K_{n1} , SELECT(node_id $_N$)); 38. Generate node_id $_T$ = H(offerIP, K_{n1}); 39. if SignG == SignN and node_id $_T$ == node_id $_N$ then 40. Generate SignP (K_{p2} , ACK); 41. Send (ACK + SignP) message to N_n ; 42. Send (NextIP + SignP) to other nodes having the same x.y.j values; 43. else <i>// unauthenticated node</i> 44. Drop the (SELECT(node_id $_N$) + SignN) message; 45. exit ; |
|--|--|

Table 3 Algorithm 3: unique IP address generation for new node N_n . generate_IP(my_IP)

| | |
|--|---|
| 1. begin 2. get my_IP $\leftarrow x.y.j.i$; Set static count $\leftarrow 1$; 3. Static J $\leftarrow 0$; Static Y $\leftarrow 0$; Static NextIP = 2; 4. if y = 0 and j = 0 then 5. J = J + 1; 6. if J ≤ 254 then 7. return NEWIP x.y.J.i; 8. end 9. else if y = 0 and j != 0 then 10. Y = Y + 1; 11. if Y ≤ 254 then 12. return NEWIP x.Y.j.i; 13. end | 14. else if y != 0 and j = 0 then 15. J = J + 1; 16. if J ≤ 254 then 17. return NEWIP x.y.J.i; 18. end 19. else if y != 0 and j != 0 then 20. i = NextIP; 21. NextIP ++; 22. if i ≤ 254 then 23. return NEWIP x.y.j.i; 24. else 25. IP address is not available; 26. end |
|--|---|

Table 4 Algorithm 4: *partition handler*

| | |
|--|--|
| 1. begin 2. Set $my_ip \leftarrow x.y.j.i$; 3. Set $my_nid \leftarrow node_id_x$; 4. Set $my_MID \leftarrow MID_x$; 5. if HELLO () message is received from other partition with MID_y then 6. if $MID_x < MID_y$ then 7. if (number of neighbor of node (child node) with MID_x) < (number of neighbor of node (child node) with MID_y) then | 8. $config = false$; 9. call the address allocation algorithm as the new node N_n 10. else 11. Set $my_MID \leftarrow MID_y$; 12. else 13. Set $my_MID \leftarrow$ generate new MID_x which is greater than MID_x ; 14. end |
|--|--|

information of the IP of the left node (i.e., when REPLY to the HELLO message is not received from any node that reflects that the node has left the MANET).

Partition and merging in MANET: Graceless/graceful departure or mobility of MANET leads to network partitioning. So, it generates another MID for it and broadcasts to the neighbors in the new range and make a new MANET network but with same IP address.

Each node in MANET is uniquely identified by a tuple which consists of (IP address, Node_id, MID). Then, the MID is checked, and if it is different, then partition handler (*Algorithm 4*) algorithm is followed. Let a MANET with MID_x gets a HELLO () message from the other partition having MID_y . If the MID_x is lesser than the value of MID_y , then the number of nodes in each MANET is compared. The MANET having lesser number of existing nodes will configure the nodes in it using address allocation algorithm or else set the MID with greater value (Table 4).

4 Conclusion

In our scheme, each node acts as proxy node capable of allocating and generating unique IP for a new node. So the DAD is not required. The calculation cost and overhead of each node are decreased as the highest priority proxy node only sends the unique IP address to the new node. By applying time stamp, arrival conflict is diminished. Every mobile node in MANET should maintain a table of IP address, status of IP address (allocated, free, father), and public key of allocated node in a MANET. This table increases the accessibility of MANET as each node is aware of state of the MANET nodes. This is indeed a low-cost addressing scheme and authenticated too. The security threats are avoided by the proposed authentication scheme. The authenticated node can only get the IP address while joining the network.

References

1. Vaidya, N.H.: Weak duplicate address detection in Mobile Ad Hoc Networks. In: Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc02), pp. 206–216 (2002)
2. Weniger, K.: Passive duplicate address detection in Mobile Ad Hoc Networks. In: WCNC, (Florence, Italy) (2003)
3. Al-Mistarihi, M., Al-Shurman, M., Qudimat, A.: Tree based dynamic address auto configuration in mobile ad hoc networks. In: Elsevier, Computer Networks, vol. 55, pp. 1894–1908 (2011)
4. Ghosh, U., Datta, R.: A secure dynamic IP configuration scheme for mobile ad hoc networks, Ad Hoc Networks. In: Elsevier Journal Published, vol. 9(7), pp. 1327–1342 (2011)

Intelligent Computing, Communication and Devices

Proceedings of ICCD 2014, Volume 2

Jain, L.C.; Patnaik, S.; Ichalkaranje, N. (Eds.)

2015, XX, 535 p. 249 illus., Softcover

ISBN: 978-81-322-2008-4