RAA: A Ring-Based Address Autoconfiguration Protocol in Mobile Ad Hoc Networks

Yuh-Shyan Chen and Shih-Min Lin

Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 621, Taiwan

Abstract. The problem for dynamic IP address assignment is manifest in mobile ad hoc networks (MANETs), especially in 4G all-IP-based heterogeneous networks. In this paper, we propose a ring-based address autoconfiguration protocol to configure node addresses. This work aims at the decentralized ring-based address autoconfiguration (DRAA) protocol, which has the capability to perform low latency and whose broadcast messages are reduced to lower control overhead. In addition, we introduce the centralized ring-based address autoconfiguration (CRAA) protocol to largely diminish control overhead and to serve as an even solution for IP address resource distribution. Both of DRAA and CRAA protocols are low-latency solutions because each node independently allocates partial IP addresses and does not need to perform the duplicate addresses detection (DAD) during the node-join operation. Communication overhead is significantly lessened in that RAA (DRAA and CRAA) protocols use the logical ring, thus utilizing fewer control messages solely by means of uni-cast messages to distribute address resources and to retrieve invalid addresses. Especially, the CRAA protocol reduces larger numbers of broadcast messages during network merging. The other important contribution is that our CRAA protocol also has an even capability so that address resources can be evenly distributed in each node in networks; this accounts for the reason our solution is suitable for large-scale networks. Finally, the performance analysis illustrates performance achievements of RAA protocols.

Keywords: Autoconfiguration, IP address assignment, MANET, RAA, wireless IP.

1 Introduction

Multiple functions of the fourth-generation (4G) communication system are envisioned to be extensively used in the near future. 4G networks are an all-IP-based heterogeneous network, exploiting IP-based technologies to achieve integration among multiple access network systems, such as 4G core networks, 3G core networks, wireless local area networks (WLANs) and MANETs. In IP-based MANETs, users communicate with others without infrastructures and service charges. A MANET is made up of identical mobile nodes, each node with a limited wireless transmission range to communicate with neighboring nodes. In

X. Jia, J. Wu, and Y. He (Eds.): MSN 2005, LNCS 3794, pp. 773–782, 2005.

order to link nodes through more than one hop, multi-hop routing protocols - such as DSDV, AODV, DSR, ZRP and OLSR - are designed. These multi-hop routing protocols require each node to have its own unique IP address to transmit packets hop by hop toward the destination. Hence to practice these routing protocols in a correct manner, a node must possess an IP address bearing no similarity to that of any other node.

In recent years, many solutions for dynamic IP address assignment in MANETs have been brought out. According to their dynamic addressing mechanisms, we organize these solutions into the following four categories: all agreement approaches [1][3][5], leader-based approaches [2][9], best-effort approaches [8] [10] and buddy system approaches [4][7]. The all agreement approach [5] featured a distributed, dynamic host configuration protocol for address assignment called MANETconf. The greatest communication overhead will be produced during network merging because nodes in networks must perform DAD. Among leader-based approaches, Y. Sun *et al.* [2] proposed a Dynamic Address Configuration Protocol (DACP). The biggest drawback Leader-based approaches bear is that the workload of the leader node is too heavy due to DAD for all joining nodes. Among best-effort approaches, the prophet address allocation protocol [10] makes use of an integer sequence consisting of random numbers through the stateful function $f(n)$ for conflict-free allocation. Prophet does not perform DAD to reduce communication overhead during network merging, but nodes with the smaller network identifier (NID) change their IP addresses no matter duplication occurs or not. Although Prophet brings the benefit of lower communication overhead, nodes break all on-going connections with the smaller NID. When two large networks merge, the impact of connection loss is significant. A.P. Tayal *et al.* [7] proposed an address assignment for the automatic configuration (named AAAC), to which the buddy system is applied whenever resources run out and new nodes seek to join a network. In AAAC, every node in merging networks broadcasts its IP address and address pool to whole networks for network merging, whose communication overhead is large.

To offer effective IP address assignment in a dynamic network environment, the solution provided by our addressing protocol presents three goals, aiming at efficient, rapid IP address distribution as well as applicable address resource maintenance: 1) low latency, 2) low communication overhead and 3) evenness. Namely, low latency produces the results that a requested node timely gets a unique address in the IP address assignment process, communication overhead is lessened to enhance network efficiency, and address resources are evenly distributed in each node.

The remaining sections of the paper are organized as follows. Section 2 proposes protocol comparisons and basic ideas of RAA protocols. In Section 3, we present the details of the decentralized RAA protocol (DRAA) with regard to address resource maintenance and node behavior handling. In Section 4, the centralized RAA protocol (CRAA) is introduced. Section 5 shows the performance analysis. Finally, Section 6 draws conclusions for the paper.

2 Basic Ideas

In this section, we proffer both conceptual discrepancies among various protocols and the basic idea of RAA protocols. The key to determining the effectiveness of a dynamic IP address assignment protocol mainly lies in the latency of node joining, and we illustrate the comparison among various protocols in terms of node joining in Fig. 1. We consider mobile wireless networks where all nodes use IP address to communicate with others. Such a network can be modeled as follows. A mobile wireless network is represented as a graph $G = (V, E)$ where V is the set of nodes and E is the edge set which gives available communications. In a given graph $G = (V, E)$, we denote by $n = |V|$ the number of nodes in the network. The identifier (ID) and the related list of node u are represented as N_u and RL_u respectively. The network identifier is represented as NID, which is 2-tuple: <IP address, Random number>, by whose uniqueness is distinguished in different networks. The ID of a node is 2-tuple: <NID, IP address>. The successor, the predecessor and the second predecessor of node u are represented as S_u, P_u and SP_u.

In Fig. 1(a) and (b), MANETconf and DACP are not conflicting free protocols, so they have to perform DAD during node joining, which increases the latency of node joining. On the contrary, the other protocols, such as Prophet, AAAC, DRAA and CRAA, are conflicting ones so that they can assign a unique address to new nodes without DAD. Furthermore, AAAC, DRAA and CRAA are categorized into buddy system approaches, whose main difference is the evenness of address blocks in all nodes during node joining. In AAAC and DRAA, a new node N_j broadcasts a one-hop address request (AREQ) to its neighbors and awaits the very first address reply (AREP). As Fig. 1(e) shown, the CRAA protocol awaits AREPs of all neighbors and picks up the biggest address block for use, which effectively improves the uneven distribution of address blocks shown in the former case.

In this paper, we draw on a novel technique developed in peer-to-peer (P2P) networks to provide a logical view for resource maintenance. It offers a logical network, allowing clients to share files in a peer-to-peer way. One of the distributed hash table (DHT) based approaches in P2P, known as Chord [6],

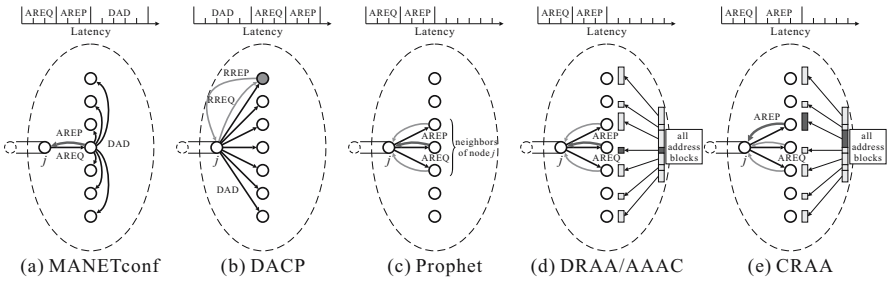

Fig. 1. The comparison of various protocols during node joining

inspires us to utilize its logical view to perform address resource management. In the prototype of Chord, in order to keep load balancing, each node uses the hash function to produce a node identifier. Unlike Chord, our solutions - RAA protocols - are not necessary to use any hash function to distribute address resources because IP address resources differ from file ones. If the hash function is applied to distribute address resources for the evenness in networks, the complexity of address management will be raised and not be suitable in MANETs. For this reason, the buddy system is combined in RAA protocols to manage resources. The binary buddy system is one common, famous type of buddy systems with a resource size of 2^m units and starting off this size. When the application issues a request, a 2^m-unit block is split into two with 2^{m-1} units respectively. However, resources in the binary buddy system are depleted rapidly. When a node owns 2^m addresses and allocates them to other nodes up to m times, address resources will be consumed. In the traditional buddy system approach (AAAC), address resources are retrieved solely during resource consumption. If a network changes with frequency, then resources usually run out. If so, the latency of a new node's requiring an available address block inevitably increases. In our solution, address resources are retrieved during both resource consumption and node joining.

In RAA protocols, each node records its logical neighbors' IDs on the related list (RL). Logical neighbors are the successor, the predecessor and the second predecessor. Notice that the RL is updated during node joining and network merging. If N_i exists in RAA protocols, the successor (S_i) is the node which allocates an address block to it and is also the first node in the clockwise course starting from N_i. The predecessor (P_i) and the second predecessor (SP_i) are the first node and the second node in the anticlockwise course starting from N_i respectively. The successor's, the predecessor's and the second predecessor's IDs can be represented as SID_i, PID_i and $SPID_i$. The RL of N_i is RL_i: $\{SID_i$, PID_i, $SPID_i\}$. Each node possesses its own anticlockwise-course free address block. During node leaving, only the successor in the clockwise course has to be informed; then it will retrieve the address block of the leaving node. Our solution achieves highly-efficient resource management and address allocation through the combination of logical ring and the buddy system.

3 Decentralized Ring-Based Address Autoconfiguration Protocol (DRAA)

Since nodes in MANETs have joining, leaving, partitioning and merging behaviors, this section sheds light on how the DRAA protocol handles these behaviors. Before we introduce it, the state diagram of RAA protocols should be shown first to help handle the node behaviors. There are totally five states in RAA protocols: INITIAL, STABLE, MERGE, HS and FINISH. When a new node intends to join a network, it enters the INITIAL state and waits for a free address block. A node enters the STABLE state when getting an address. If nodes are informed of network merging by some node, they enter the MERGE state for merging. If the holder leaves the network, other nodes enter the HS (holder selection) state

to select a new holder. If leaving the network, a node enters the FINISH state. According to node behaviors, we realize how the DRAA protocol deal with joining and leaving. In following paragraphs, we will state the initiation of networks and specify node behaviors via the DRAA protocol.

3.1 Initial State

A node is in the INITIAL state before getting a usable address. At the start, the first node, N_i, enters a network and broadcasts a one-hop address request (AREQ) message. N_i awaits an address request timer (AREQ_Timer) to gather responses from other nodes. Once the timer expires and N_i does not get any response, N_i gets its identity as the first node (i.e. the holder) in the network. N_i randomly chooses an IP address, and sets it into its ID. The holder uses its ID and a random number as the network identifier (NID). The NID is periodically broadcast to the whole network by the holder, enabling nodes to get the NID of their locating network and to detect network partitioning and mergeing.

3.2 Node Joining

When aiming to enter a network, N_j needs to obtain an IP address and then checks whether P_j exists in the network in order to ensure the wholeness of address blocks. The joining procedure consists of two phases: 1) address requesting and 2) failed node checking. The address requesting phase is used to allocate an IP address to a new node whereas the failed node checking is used to check whether the predecessor of the new node is alive.

Address Requesting Phase. With N_j in the initial state and intending to join the network in DRAA, the address requesting phase will be triggered. The address requesting phase is applied to allocate an IP address and an address block to a new node, including an address request (AREQ) issued from the new node, an address reply (AREP) replied from neighbors of the new node and an address reply acknowledgement (AREP_ACK) sent by the new node to the neighboring node which is the first to transmit an AREP.

Failed Node Checking Phase. After the address requesting phase, N_j uses the IP address to communicate with other nodes in the MANET and enters the STABLE state from the INITIAL state. The Failed node checking phase is used to check whether the predecessor of the new node is alive. The new node first sends an alive checking (ACHK) message to its predecessor. If the predecessor alive, the predecessor replies with an ALIVE message to the new node. If the predecessor fails, the new node sends an address retrieve (ARET) message to its second predecessor to retrieve the address block of the predecessor. Then the second predecessor sends back an address retrieve acknowledgement (ARET_ACK) to the new node to inform it which ones are its new predecessor and second predecessor.

3.3 Node Leaving

When a node leaves the network genteelly, the leaving node sends a LEAVE message to its successor, and then the successor takes over the address block of the leaving node. Notice that during node leaving, the successor is unnecessary to be the one-hop neighbor of the leaving node, which has to procure the successor's routing path in the routing table. If nodes crash without a LEAVE message, the orphan block of the crashed node is retrieved in the failed node checking phase.

3.4 Network Partitioning

The only responsibility of the holder node in the DRAA protocol is to broadcast its NID. If there is a node not receiving the NID after one broadcasting period, the node may have been partitioned from the network. But wireless communication is not reliable; the NID may not be received due to packet loss. To reduce the impact, we define network partitioning as three broadcasting periods without the need to receive the NID. Once a node detects network partitioning, it performs the holder selection algorithm.

Holder Selection. In the holder selection algorithm, we borrow the random backoff in carrier sense multiple access with collision avoidance (CSMA/CA) protocol of IEEE 802.11 standard and slightly modify the idea. When network partitioning is detected by the node, the detecting node enters the holder selection (HS) state and chooses a random backoff timer (RB_Timer) which determines the amount of time the node must wait until it is allowed to broadcast its NID. The RB_Timer reduces the collision of NID messages, broadcast by nodes in the holder selection state.

3.5 Network Merging

Assume that there are two networks G (NID_G) and G' ($\mathrm{NID}_{G'}$) intend to merge. Network G has n nodes and network G' has n' nodes. In the DRAA protocol, all nodes during network merging have to broadcast its ID for DAD. When all nodes get full information of networks, they choose a holder who broadcasts a bigger NID. If duplicate addressing occurs, duplicate nodes which have fewer TCP connections or are in the smaller network should rejoin the network. The benefits of choosing the node from a smaller network are quicker responses and lesser possible disconnections.

4 Centralized Ring-Based Address Autoconfiguration Protocol (CRAA)

With regard to low communication overhead and evenness, the centralized RAA protocol (named CRAA) is introduced. The differences between DRAA and CRAA protocols are in node joining and network merging. In the address requesting phase of node joining, the CRAA protocol selects the biggest free address block to use during the address requesting phase. Furthermore, the CRAA

protocol in the failed node checking phase can recover more than one failed node because the holder in CRAA maintains the node list (NL) which contains all used IP addresses in the network. The NL is helpful when there are two or more continuous node failures in a ring, so that the lost address resources can be retrieved with the help of the holder. During networks merging, the NL reduces broadcast messages by exchanging the holder's NL only.

In the address requesting phase, the new node in the CRAA protocol waits for all neighboring nodes' AREP messages and selects the biggest free address block for use. The DRAA protocol can distribute valid addresses immediately, but the address block of each node will probably not be evenly for the long-term perspective. The advantage of the CRAA protocol is averaging the number of free address blocks managed by each node. Since the holder maintains the NL of the network in the CRAA protocol, each node sends an address registration request to the holder when a node takes over a free address block. After using a new node uses a new IP address, the new node sends a registration request (RREQ) to the holder that records the new node's address on the NL and the holder sends a registration reply (RREP) to the new node.

By contrast with the DRAA protocol during network merging, in the CRAA protocol, when the holder receives the merging request (MREQ), it broadcasts its own NL. Each node receives the holders' NLs, merges those NLs, modifies its own RL and chooses the bigger NID to be its network identifier. This method reduces large communication overhead during network merging. The NL becomes much significant in CRAA, so that we will introduce the replication of the NL. The NL incorporates in all used nodes in the network and is a helpful information source for both the lost address block retrieval and network merging. The NL replicas exist in the holder's successor, predecessor and second predecessor. When the holder gets an RREQ message, the holder copies the newest NL to these three nodes to ensure the freshness of the NL.

5 Performance Analysis

To make appropriate protocol simulation, all protocols are implemented with C++. We select AAAC [7] and Prophet [10] to compare with our protocols, and do not consider comparing with MANETconf and DACP. Both of AAAC, Prophet are conflicting free protocols, while MANETconf and DACP are not. The simulation parameter is described below. The simulation time is 1500 seconds. The speed of nodes is varied from 0 to 10 m/s, and the pause time is set to 0 for strict reasons. The mobile nodes move according to the random waypoint mobility model. The network size is set to 1000m $\times$ 1000m where the number of nodes is from 50 to 300. The size of address blocks is set to 2^m IP addresses and m are 10 and 24. The maximal size of the address block is 24-bit which makes a suitable selection for large networks. In particular, in our simulation, the transmission range is 250m. The underlying routing protocol applies the DSDV in all simulations. The network is initialized with a single node. Nodes join the network every 1 second and arriving nodes are placed randomly in the

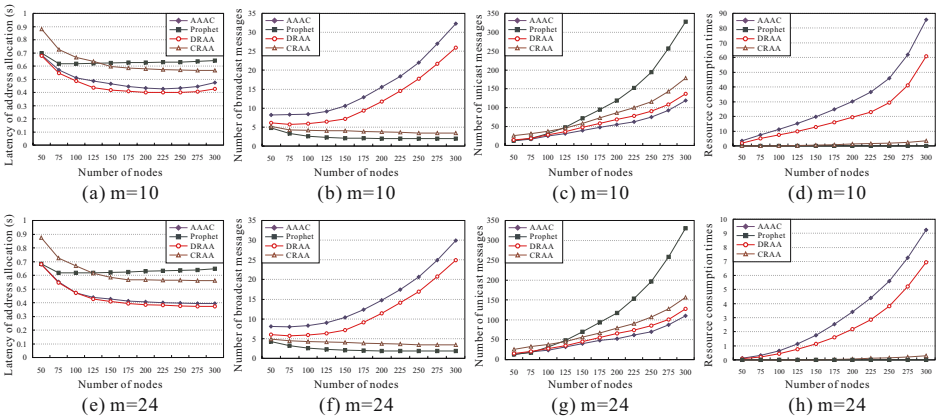

Fig. 2. Average latency of address allocation, the number of broadcast/unicast messages and resource consumption times with the varied number of nodes at m=10 and 24

rectangular region. The beacon interval for the holder to broadcast its NID is set to 10 seconds. The maximum retry times r of AREQ is set to 3 [10], and the AREQ_Timer is set to 2 seconds for quicker addressing. For strict reasons, when all nodes are stable, node leaving and rejoining, network partitioning and network merging are produced freely. The performance metrics of the simulation are given below.

- *Latency*: Latency of address allocation represents the average latency for a new node to obtain a unique IP address within the network. The shorter the latency, the better, since it means a new node can get a usable IP address more rapidly.
- *Communication overhead*: Communication overhead refers to the number of control messages transmitted during the simulation period, including unicast and broadcast messages. Normally, broadcast messages occupy more bandwidth than unicast messages do.
- *Evenness*: Evenness implies that address blocks should be evenly distributed in all nodes, which indicates that each node has the capability to assign address blocks to newly-joined nodes. The more evenly address blocks are allotted, the fewer the address resource consumption times in each node are, from which we are able to determine whether address blocks are evenly distributed.

Fig. 2(a) and (e) depict the average latency of address allocation versus the varied number of nodes, with two address block sizes: 2^{10} and 2^{24} respectively. In general, the DRAA protocol has the shortest latency because the DRAA protocol fast replies an address request and maintains orphan address blocks when a node joins the network. The CRAA protocol has longer latency than AAAC and DRAA protocols because it awaits AREPs of neighbors to select a

bigger address block. Prophet has the longest latency when the number of nodes is more than 150 because during network merging, all nodes in the smaller NID have to rejoin the network and wait for the expiration of an AREQ_Timer. When the number of nodes is 50, all protocols have longer latency because the network size is 1000m × 1000m and the transmission range is 250m. When a new node joins the network, it has the higher probability that no node is within its transmission range, so that the new node needs to await the AREQ_Timer expiration. When the number of nodes is more than 150 at m=10 in Fig. 2(a), the latency of AAAC and DRAA increases with the node number because more resource consumption times in the whole network will expand the number of address requesting phase retries.

Fig. 2(b) and (f) show the number of broadcast messages versus the varied number of nodes, with two address block sizes: 2^{10} and 2^{24} respectively. Prophet has the fewest broadcast messages because it does not preform DAD during merging and all nodes with the smaller NID have to rejoin the network. The CRAA protocol has the second fewest broadcast messages because only the holder broadcasts NL for DAD during network merging. The DRAA protocol has fewer broadcast message than AAAC because the resource consumption times of DRAA are less than AAAC. Fig. 2(c) and (g) display the number of unicast messages versus the varied number of nodes, with two address block sizes: 2^{10} and 2^{24} respectively. AAAC has the fewest unicast messages because most of its control messages are handled with broadcast messages. In the CRAA protocol, more unicast messages are added to maintain the NL, so its unicast messages are more than those in DRAA as well as AAAC. Unicast messages in Prophet are the most because many nodes need to rejoin the network during network merging, which results in the large number of unicast messages.

Fig. 2(d) and (h) show the resource consumption times versus the varied number of nodes, with two address block sizes: 2^{10} and 2^{24} respectively. The DRAA protocol, during node joining, has the failed node checking phase to retrieve orphan blocks, so the resource consumption times decrease. The CRAA protocol has evenly-distributed resources because nodes in the CRAA protocol request address blocks from all neighbors and choose the biggest one to use. Although the way Prophet allots addresses is different from that of buddy system approaches and does not guarantee every allotted address is unique, $f(n)$, which distributes addresses, does not have the phenomenon of resource consumption. To be fair, the resource consumption times of Prophet are set to be 0. On the whole, when the address block size is big enough, the resource consumption times will be significantly reduced and it is the CRAA protocol whose resource consumption times are close to be 0.

6 Conclusions

This paper proposes two ring-based address autoconfiguration protocols in mobile ad hoc networks. Compared with existing address assignment protocols, the DRAA protocol successfully achieves low latency and low communication

overhead, and the CRAA protocol further achieves low communication overhead and evenness of dynamic address assignment. RAA protocols use a logical ring to proceed address allocation and resource management. The ring provides unique address assignment without DAD. The DRAA protocol tolerates one node's invalidity and restores a failed node without help of the holder, and the CRAA protocol restores failed nodes with help of the holder. Based on the above advantages, RAA protocols shows high efficiency in address allocation as well as in resource management and suitability for the large scale mobile ad hoc network.

References

1. C. E. Perkins, J. T. Malinen, R.Wakikawa, E. M. Belding-Royer, and Y. Sun. Ad hoc Address Autoconfiguration. IETF Internet Draft, draft-ietf-manet-autoconf-01.txt, 2001. (Work in Progress).
2. Y. Sun and E. M. Belding-Royer. Dynamic Address Configuration in Mobile Ad hoc Networks. Technical report, Computer Science Department, UCSB, Mar. 2003.
3. Zero Configuration Networking. http://www.zeroconf.org/.
4. Mansoor Mohsin and Ravi Prakash. IP address assignment in a mobile ad hoc network. In *Proceedings of IEEE Military Communications Conference (MILCOM 2002)*, volume 2, pages 856–861, Anaheim, CA, United States, 7-10 Oct. 2002.
5. Sanket Nesargi and Ravi Prakash. MANETconf: Configuration of hosts in a mobile ad hoc network. In *Proceedings of the Twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, volume 2, pages 1059–1068, 23-27 Jun. 2002.
6. I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, pages 149–160, 2002.
7. Abhishek Prakash Tayal and L. M. Patnaik. An address assignment for the automatic configuration of mobile ad hoc networks. *Personal Ubiquitous Computer*, volume 8, issue 1, pages 47–54, Feb. 2004.
8. Nitin 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 (MobiHoc 2002)*, pages 206–216, Lausanne, Switzerland, 9-11 Jun. 2002.
9. Kilian Weniger and Martina Zitterbart. IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. In *Proceedings of European Wireless 2002*, pages 142–148, Florence, Italy, Feb. 2002.
10. Hongbo Zhou, Lionel M. Ni, and Matt W. Mutka. Prophet address allocation for large scale MANETs. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, volume 2, pages 1304–1311, San Francisco, CA, United States, Mar. 30-Apr. 3 2003.