

Zone Routing Determination for IZRP Based on Bee-inspired Algorithm

Tung Thanh Le

Dept. of IT Convergence Engineering
Kumoh National Institute of Technology
Gumi, Gyeongbuk 730–701, Korea
Email: ttungl@kumoh.ac.kr

Do Duy Tan

Dept. of IT Convergence Engineering
Kumoh National Institute of Technology
Gumi, Gyeongbuk 730–701, Korea
Email: tandd@kumoh.ac.kr

Dong-Sung Kim

Dept. of IT Convergence Engineering
Kumoh National Institute of Technology
Gumi, Gyeongbuk 730–701, Korea
Email: dskim@kumoh.ac.kr

Abstract—In this paper, a new zone radius determination algorithm, based on an adaptive nature-inspired routing protocol that emulates the foraging behavior of bees and their ability to find an optimal route from the hive to the nectar sites, is proposed. Instead of tuning one-hop-by-one-hop radius of nodes as in IZRP, our proposed algorithm uses the difference zone radii between adjacent nodes to calibrate the zone radius to adapt quickly the network conditions. Through simulation results, we compared the execution time of two zone-routing algorithms under different network scenarios. The simulation results proved the efficiency of our proposed algorithm in reducing the control traffic overhead and increasing the throughput.

Keywords—zone radius; independent zone routing protocol; hybrid routing protocols; mobile ad hoc networks (MANETs); bee-inspired algorithm.

I. INTRODUCTION

Nowadays, there is no skeptical that mobile ad hoc networks can be deployed quickly to provide robust communication in a variety of highly dynamic environments. This feature makes them extremely suitable for a wide range of fields such as supporting for tactical communication in the military, emergency response efforts, civilian areas such as convention centers, construction sites [1], and geographical areas prone to natural disasters. Thus, mobile ad hoc networks can be envisaged to operate over a wide range of coverage regions with varying node dispersals, node densities, or mobility targets under varying network conditions [2] [3] [4].

The zone routing protocol (ZRP), also known as a hybrid routing protocol, offers the advantage of scalability to adapt to a variety of network conditions. Typically, ZRP combines two phases, proactive routing and reactive routing. In proactive routing, ZRP proactively manages routes within a local region of network, referred to as the routing zone, by gathering information about the neighborhood. Hence, proactive routing is also called intrazone routing protocol (IARP). In contrast, reactive routing works at a global level where an on-demand routing protocol, which does not require neighborhood information, is issued [5] [6].

Network conditions in the zone routing vary drastically, and hence, ZRP requires to be dynamically reconfigured with respect to the local network conditions. The independent

zone routing protocol (IZRP) was proposed to enhance the efficiency of ZRP in the independently nodes [1] [5]. In zone routing with independently sized routing zones, each node in the network can adaptively reconfigure its own optimal zone radius in a distributed fashion. The proportion of proactive and reactive routing in this hybrid routing protocol can be fine-tuned by adjusting a single parameter – the zone radius of the node.

Typically, a zone radius (ρ) determination algorithm is based on a hybrid scheme that is a combination of the min search and adaptive traffic estimation schemes [1]. This hybrid scheme dynamically reconfigures the minimum zone radius of each node in a distributed fashion. In detailed explanation, the zone radius determination algorithm should able to determine the optimal zone radius of each node in the network as well as should be adaptively quick to any changes in the network characteristics. The purpose of this algorithm is to make a minimal amount of extra overhead network by monitoring the control traffic passing via a node, and can fine-tune to adapt to regional.

The min searching scheme iteratively is searching for the minimum of the routing traffic by increment or decrement the routing zone radius of a node by one hop. During each estimate interval, the amount of routing traffic passing via the node is measured and if the amount of routing traffic in current estimation is less than that in the previous interval, the zone radius is incremented/decremented in the same direction. The direction of zone radius change is reversed if the current estimation is greater than that in the previous interval. The process continues until a minimum is detected.

The Adaptive Traffic Estimation scheme tries to track the optimal zone radius by iteratively fine-tuning the zone radius for reducing the reactive or the proactive traffic routing dominance in the total routing overhead. The scheme lies on $\Gamma(\rho)$, which is the ratio of the reactive traffic to the proactive traffic of zone radius during a certain estimation interval, as measured at one network node. Adjustments to the zone radius are changed through the comparison of the ratio $\Gamma(\rho)$ with a predetermined threshold, Γ_{thres} . If $\Gamma(\rho) > \Gamma_{thres}$, ρ is incremented by one-hop to decrease the reactive traffic routing, otherwise ρ is decremented by one-hop to decrease the proactive traffic routing. However, the change of zone

This work was supported in part by the National Research Foundation of Korea (NRF) under Grant #2011-0025409.

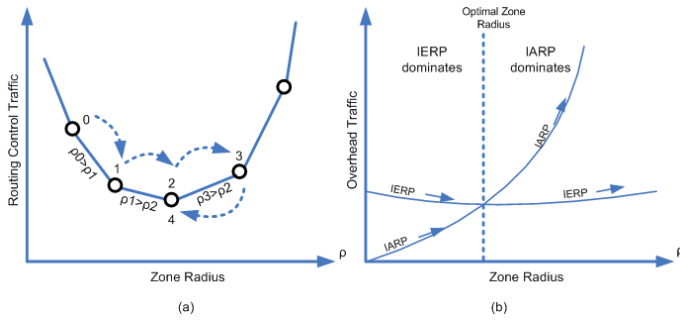


Fig. 1. (a) min searching algorithm and (b) zone routing radius optimization of ZRP.

radius after each estimation interval probably could lead to the unstably network.

In fact, there are many papers which have been concerned about ZRP and IZRP with comparison of the performance analysis in its characteristics to that of other routing protocols as in [7] [8] [9]. However, there is just few papers involving the bio-inspired behaviors that could probably bring a lot of interesting about amazing characteristics of our nature, in which is not exploited yet. For example, in [10], the authors proposed a combination between ZRP and Ant Colony Optimization (ACO) for their proposed protocol, is called Ant-based Dynamic Zone Routing (AD-ZRP), to improve the quality of dynamic network conditions. Thus, it is obvious that bio-inspired behaviors need to be applied into wireless communications for enhancing the performance efficiency.

In this paper, we propose a zone radius determination algorithm that emulates the characteristics of bees while foraging for nectar. Further, through a simulation, we demonstrate that the computational complexity of the proposed algorithm is lower than that of the IZRP zone radius determination algorithm. The IZRP algorithm is only incremented by one hop in the min search scheme, and hence, it cannot determine a new optimal zone radius instantaneously. This problem leads to the hysteresis for adaptation in the network. However, as described later, in the algorithm based on bees foraging for nectar, onlooker bees O_b can determine profitability of nectar sites by comparing them with previous one in memory. Similarly, in our proposed algorithm, the zone radius of a node is compared with that of its adjacent neighbor node, and then, the node sends a message requesting its neighbor to change the zone radius difference. The simulation is performed to compare the computational complexity of both algorithms in terms of their execution time. The simulation results prove that our proposed algorithm can reduce the control traffic overhead and increase the throughputs of the network. Further, our algorithm shows the ability to rapidly adapt to network conditions because of its low computational complexity.

The rest of the paper is organized as follows. We issue the problem formulation in section II. Afterthat, a proposed algorithm is presented in section III. Simulation results and conclusions demonstrate our approaches in section IV and

section V, respectively.

II. PROBLEM FORMULATION

In this section, the problem solving performance of our proposed algorithm is compared with that of the IZRP algorithm. The parameters considered are control traffic overhead, throughput, and routing traffic overhead, as listed in Table I:

TABLE I
PROBLEM SOLVING PERFORMANCE OF IZRP AND PROPOSED ALGORITHM

Parameters	IZRP algorithm	Proposed algorithm
Control Overhead	High	Low
Throughput	Low	High
Routing Overhead	High	Low

This paper aims to achieve the following objectives:

- Reduce the control traffic overhead and increase the throughputs of the routing zones.
- Reduce the hysteresis for adaptation in the independent routing zones.
- Decrease the computational complexity of the network load.

III. BEE-INSPIRED ALGORITHM

A. From nature to network communications

The nature-inspired artificial bee colony (ABC) algorithm is a swarm-based meta-heuristic algorithm that was introduced by Karaboga [11] [12]. In the ABC algorithm, three types of bees are considered: employed bees E_b , onlooker bees O_b , and scout bees. Typically, each cycle of the ABC algorithm is divided into three steps. (1) The employed bees are sent to possible nectar sites to measure the profitability (fitness values) of these nectar sites. (2) The onlooker bees receive this information, which is shared by the employed bees through a waggle dance [13], as shown in Fig. 2. (3) Scout bees are selected and sent to the nectar sites.

In [13], two nectar sites having different quality of nectar were considered to demonstrate the recruitment of bees for foraging, as shown in Fig. 2. In step 1, after the employed bees identify the nectar sites, they fly back to the hive and begin a waggle dance. They dance with different vibrations depending on the profitability of the nectar sites to attract the onlookers¹. After a couple of hours, as shown in step 2, many onlooker bees following the dance of an employed bee that has found a better nectar site. Several hours later, in step 3, the dance of the employed bee that has found the better nectar site is followed by a majority of the onlookers; then, the scouts perform the process of exploring the better site.

Let i be the set of nectar sites ($i = 1, 2, \dots, E_b$), E_b be the number of employed bees, and F_i be the fitness value of a site i . The probability p_i that onlooker bees select a nectar site is given as follows:

¹In Fig. 2, the blue nodes are the number of the onlookers those follow to the better nectar site, meanwhile the red nodes are the number of the onlookers those follow to the poorer nectar site.

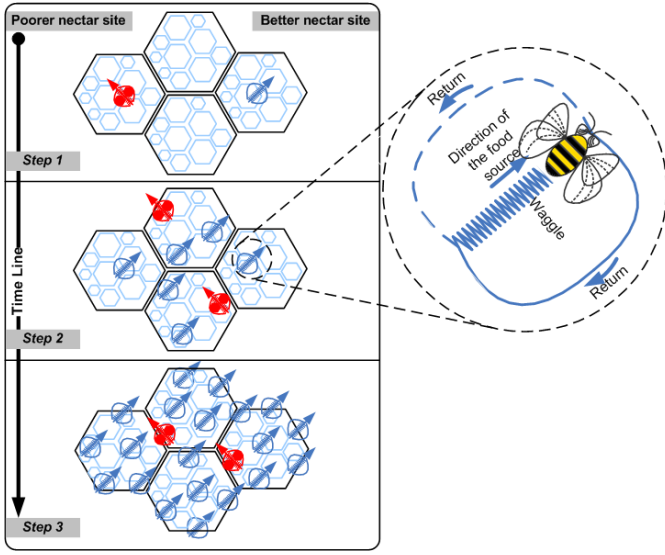


Fig. 2. Diagram showing recruitment of bees for foraging; *Step 1*: Employed bees begin a waggle dance after finding nectar sites; *Step 2*: Many onlooker bees follow the waggle dance of the employed bees; *Step 3*: The dance of bees that have found a better nectar site is followed by a large number of onlookers; then, the scouts perform the process of exploring the better nectar site.

$$p_i = \frac{F_i}{\sum_{i=1}^{E_b} F_i}. \quad (1)$$

According to this equation, a good nectar site will attract more onlooker bees than a bad one. After the onlooker bees select a nectar site i , they consider a nectar site $i + 1$ in the neighborhood of the selected one i and compare the amount of nectar in $i + 1$ with that in i . Once all onlookers have chosen their sites, each onlooker calculates the profitability of the new nectar site. The new location is memorized and the previous one is discarded if the profitability of the new site is better than that of the previous one; if not, the previous site is retained in memory. The process is repeated until termination criteria are met.

To determine the location of a neighboring nectar site with respect to the previous one in memory, the ABC algorithm changes one randomly chosen parameter and keeps the other parameters constant. The location of the nectar site is determined as follows:

$$S_{ij}^{new} = S_{ij}^{old} + \sigma(S_{ij}^{old} - S_{kj}), \quad (2)$$

where σ is uniformly distributed between $[-1, 1]$ and $j \in \{1, 2, \dots, D\}$, in which D is the dimension of the solution vector; $k \neq i$.

After a nectar site has been abandoned, the employed bee associated with it becomes a scout. The scout generates a new nectar site location as follows:

$$S_i^{j(new)} = MinS_i^j + Rand(0, 1)(MaxS_i^j - MinS_i^j), \quad (3)$$

where $S_i^{j(new)}$ applies to all j parameters and $Rand(0, 1)$ is randomly generated by the scout.

Thus, it is obvious that the ABC algorithm is based on the following parameters: (1) the colony size contains E_b and O_b ; (2) the limit value, which is the maximum number of routes to the nectar sites, and (3) the maximum cycle number.

Algorithm 1 Artificial Bee Colony Algorithm

- 1: **procedure** ABC ALGORITHM(S_i)
 - 2: **Input:** S_i is the solutions (food source locations) of the size of population with ($i = 1, 2, \dots, E_b$);
 - 3: **Output:** $cycle = MCN \rightarrow$ **Optimal solution.**
 - 4: $cycle = 1$;
 - 5: Calculate the fitness values of these solutions.
 - 6: Adjust S_i locations from solutions' neighbors.
 - 7: Calculate the fitness values of adjusted locations.
 - 8: Compare food source locations and retained best.
 - 9: **if** O_b cycle: $O_b^{curr} = O_b^{prev}$ **then**
 - 10: Approached limit value condition.
 - 11: **else**
 - 12: Calculating the p_i values of S_i locations.
 - 13: **end if**
 - 14: Determine S_i location corresponds to high p_i value.
 - 15: Adjust the determined location S_i
 - 16: Count O_b
 - 17: **if** limit value approached **then**
 - 18: Produce new food source location(s) $S_i^{j(new)}$
 - 19: Calculate the fitness value(s) $S_i^{j(new)}$
 - 20: Evaluate the result(s)
 - 21: **end if**
 - 22: Memorize the best location S_i
 - 23: **if** $cycle = MCN$ **then**
 - 24: **End: Optimal solution.**
 - 25: **else**
 - 26: $cycle = cycle + +$;
 - 27: Return: O_b cycle condition.
 - 28: **end if**
 - 29: **end procedure**
-

B. Zone Radius Determination Algorithm

A zone radius determination algorithm for ZRP has been mentioned by Haas *et al.* in [1] [3]. They assumed that IARP and interzone routing protocol (IERP) are convex functions of ρ , and hence, the total ZRP traffic is eventually convex. However, the min search scheme cannot determine a new optimal zone radius instantaneously because it is only incremented by one hop. In dense network scenarios, this drawback results in an enormous amount of control traffic in the routing zones.

The algorithm 2 shows the min search scheme and adaptive traffic estimation scheme in the IZRP zone radius determination algorithm reported by Haas *et al.*. To overcome the abovementioned disadvantage, the proposed algorithm incorporates additional messages that fine-tune the ρ of adjacent nodes directly using the difference in their zone radii, $\Delta\rho$, instead of overshooting one-hop in the min search scheme.

The inspiration for the proposed algorithm was drawn from the onlooker bees O_b that can determine fitness values by

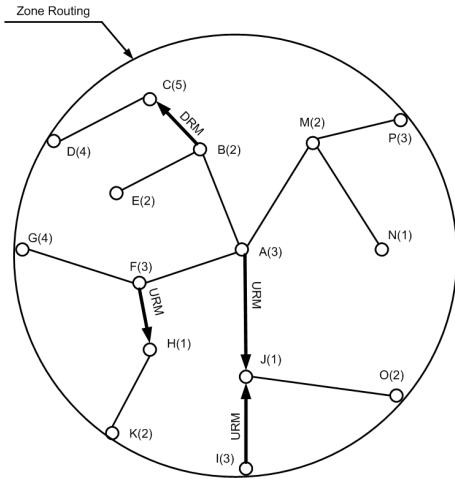


Fig. 3. Nodes request neighbors to adjust their zone radius within one-hop distance.

comparing the current nectar site with the previous one in memory. The previous site is substituted with a new one if its quality is lower than that of the new one. Thus, O_b continue the process until the termination criteria are met. In the zone radius determination algorithm, a node can compare its radius with that of its adjacent nodes in its neighborhood and request the adjacent nodes to directly change $\Delta\rho$. If the difference in the zone radius of a node and that of its adjacent neighbor node is either greater than two or less than two, then the node immediately sends an up-radius message (URM) or a down-radius message²(DRM) requesting the neighbor to scale up/down its radius on the basis of that difference, as shown in Fig. 3.

Based on the C programming language, the algorithms are performed under the direction of computational complexity in terms of the execution time in the CPU clock³ through the various input size n , which is the total control traffic corresponding to the number of hops of nodes' zone radii in the network. Let's take a look at algorithm 3, it is clear that when the number of hops in adjacent nodes are compared directly through the URM/DRM messages, the number of times to change in the second stage, which is the adaptive traffic estimation, also is reduced remarkably.

Let $T_1(n)$ be the computing time of the IZRP algorithm (Algorithm 2) for input size n . Intuitively, we assumed that the time complexity for the initialization step is 1. During the simulation, n is incremented in sets of n iteratively, and thus, the time complexity in this step is $(1 + n + (n - 1))$. The variation in the amount of IZRP traffic is based on the network medium, and hence, we assumed a uniformly distributed function between (2000, 20000) to set the input

²URM and DRM are the messages which demonstrate the number of hops in zone radii of nodes that need to be changed up or down, respectively.

³Our simulation results were performed under Intel Core2Duo 2.66Ghz (CPU), 2GB (RAM), Windows XP.

⁴ I_{A_i} and I_{E_j} are the proactive and reactive traffic component dominance in the total zone routing overhead [1].

Algorithm 2 Zone Radius Determination Algorithm in IZRP

```

1: procedure IZRP ALGORITHM
2: Input:  $n = Z(\rho)$ ; // total control traffic corresponding to
   the number of hops of nodes' zone radii in the network.
3: Output: Optimal Zone Radius ( $\rho_{optimal}$ );
4:   for Process  $n$  do
5:     Check the difference of nodes's zone radii  $\Delta\rho$ .
6:     if  $Z(\rho) < Z_{previous}(\rho)$  then
7:       for Process  $\rho_{current}$  with  $\Delta\rho$  do
8:          $\rho_{current} = \rho++$ ;
9:       end for
10:    else
11:      for Process  $\rho_{current}$  with  $\Delta\rho$  do
12:         $\rho_{current} = \rho--$ ;
13:      end for
14:    end if
15:    if  $\rho_{current} < \rho_{optimal}$  then
16:       $I_{E_j}$  increases,  $I_{A_i}$  decreases.4
17:    else
18:       $I_{A_i}$  increases,  $I_{E_j}$  decreases.
19:    end if
20:    Obtain  $\Gamma(\rho) = \frac{I_{E_j}}{I_{A_i}}$ .
21:    if  $\Gamma(\rho) > \Gamma_{threshold}(\rho)$  then
22:       $\rho_{optimal} = \rho_{current}++$ ;
23:    else
24:       $\rho_{optimal} = \rho_{current}--$ ;
25:    end if
26:  end for
27:  Return: Min Search Scheme;
28: end procedure

```

value of $Z(\rho)$, which is the current amount of IZRP traffic. The time complexity in this step is 1. Nodes rescale their zone radii on the basis of the amount of IZRP traffic; however, the rescaling is done in one-hop increments. Hence, even if the difference in the zone radii of a node and its adjacent neighbor is $\Delta\rho > 2$ or < 2 , the node is still only incremented by one hop until it meets the termination criteria. Thus, the time complexity in this step is $(1 + n + (n - 1))$ during the For loop and $(n - 1)$ during the ρ one-hop increment/decrement. Next, the algorithm enters the adaptive traffic estimation scheme stage, which determines the optimal zone radius. The current zone radius is compared with a predetermined optimal zone radius and the proactive and reactive routing behavior of the network is adjusted. Altering the zone radius after this stage could lead to instability in the network if rapid adaptation is not performed. Finally, the computing time $T_1(n)$ is estimated as follows:

$$T_1(n) = 5n + 7. \quad (4)$$

On the other hand, let $T_2(n)$ be the computing time of the proposed algorithm (Algorithm 3). The time complexity of this

⁵ $\Gamma(\rho)$ is the ratio of the IERP traffic to the IARP traffic; $\Gamma_{threshold}(\rho)$ is the ratio of the predetermined threshold [1].

Algorithm 3 Proposed algorithm to reduce overall control traffic in IZRP

```

1: procedure PROPOSED ALGORITHM
2: Input:  $n = Z(\rho)$ ; // total control traffic corresponding to
   the number of hops of nodes' zone radii in the network.
3: Output: Optimal Zone Radius ( $\rho_{optimal}$ );
4:   for Process  $n$  do
5:     if  $Z(\rho) \neq Z(\rho)_{previous}$  then
6:       Check the difference of nodes's zone radii  $\Delta\rho$ .
7:       if  $\Delta\rho = |\rho_i - \rho_j| < 2$  then
8:         node  $i$  requests the adjacent nodes to di-
9:         rectly change  $\Delta\rho$  by sending a message URM( $\Delta\rho$ );
10:        else
11:         node  $i$  requests the adjacent nodes to di-
12:         rectly change  $\Delta\rho$  by sending a message DRM( $\Delta\rho$ );
13:        end if
14:      end if
15:    if  $\rho_{current} < \rho_{optimal}$  then
16:       $I_{E_j}$  increases,  $I_{A_i}$  decreases.
17:    else
18:       $I_{A_i}$  increases,  $I_{E_j}$  decreases.
19:    end if
20:    Obtain  $\Gamma(\rho) = \frac{I_{E_j}}{I_{A_i}}$ .
21:    if  $\Gamma(\rho) > \Gamma_{threshold}(\rho)$  then5
22:       $\rho_{optimal} = \rho_{current}++$ ;
23:    else
24:       $\rho_{optimal} = \rho_{current}--$ ;
25:    end if
26:  end for
27:  Return: Min Search Scheme;
28: end procedure

```

algorithm is different from that of IZRP algorithm primarily in the min search scheme stage. Instead of fine-tuning the zone radii using one-hop increments, the difference in the zone radii $\Delta\rho$ in adjacent nodes is adjusted directly. Therefore, the computing time of the proposed algorithm, $T_2(n)$, is

$$T_2(n) = 2n + 6. \quad (5)$$

From Eq. 4 and 5, we observe that $T_2(n)$ is less than $T_1(n)$. In addition, the equations show that the size of the algorithms will grow in direct proportion to the size of the input data set. Moreover, if looking at the overall traffic in the network, the proposed algorithm can drastically reduce the enormous amount of packet transmission in nodes.

IV. SIMULATION

A. Algorithms Evaluation

In order to simulate node mobility in IZRP, we assumed all nodes to be connected. Fig. 4 shows that the average computational complexity changes slightly at low number of inputs. The average computational complexity of our proposed algorithm is lower, nearly 10% at $n = 2,000$, than that of the IZRP algorithm. When n increases, this value increases

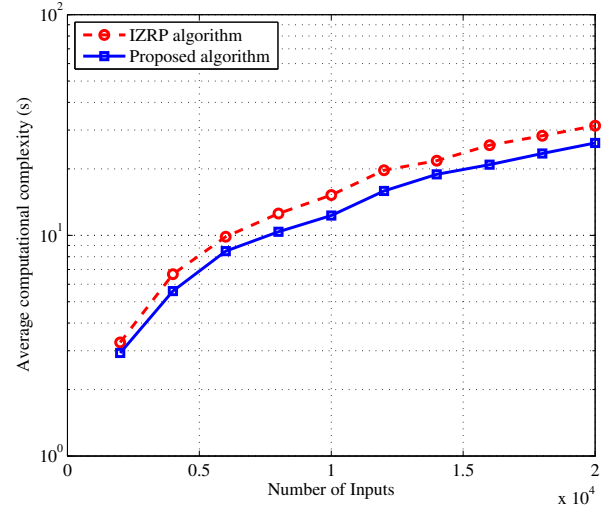


Fig. 4. Comparison results between computational complexities of zone radius determination algorithms.

to 20% and 23% at $n = 10,000$ and $20,000$, respectively. Thus, when the zone radius of the node increases, the connection ratio probably decreases because of the large network connectivity and the unpredictability of node connectivity under various network conditions. However, our proposed algorithm remarkably enhanced the connection ratio and the hysteresis for adaptation in independent zone routing because of the reduction in the average computational complexity. Logically speaking, this proposed algorithm reduces control traffic overhead and improves throughputs and the ability to adapt by altering the optimal zone radius. Therefore, the proposed algorithm can prevent a low-latency scalability of zone radius, which is the main cause of high control traffic overhead and low throughput.

B. Network Performance

The Network Simulator (NS2) simulation environment was implemented to simulate the ZRP and IZRP frameworks [14]. The network includes the various number of nodes from 5 to 30 nodes spreading randomly in the square area $300 \times 300 m^2$. A node moves at a constant speed from 0 to 30 m/s following to the Poisson distributed. The faster speed means that a node has to reduce its radius to guarantee the network connection stably, while the slower speed results in increasing the node's zone radius, as in IZRP feature. The total routing overhead is shown as the sum of IARP and IERP components.

According to the different network conditions, the simulation was implemented by different zone radius configurations. From Fig. 5, the amount of normalized traffic overhead generated during the simulation time, as 300 seconds. It can be seen that IZRP has the lowest value, with at least 40% reduction (at 5 nodes) and at least 30% reduction (at 30 nodes) in traffic, as compared to the regular ZRP. The reduction in traffic for this scenario illustrates that different zone radii in the network

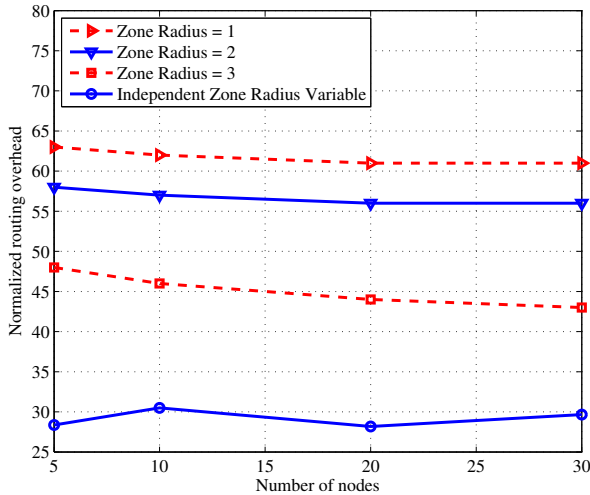


Fig. 5. Comparison results between the normalized routing overhead in ZRP with different zone radii from 1 to 3 and IZRP, as the network size is 5, 10, 20, and 30 nodes, respectively.

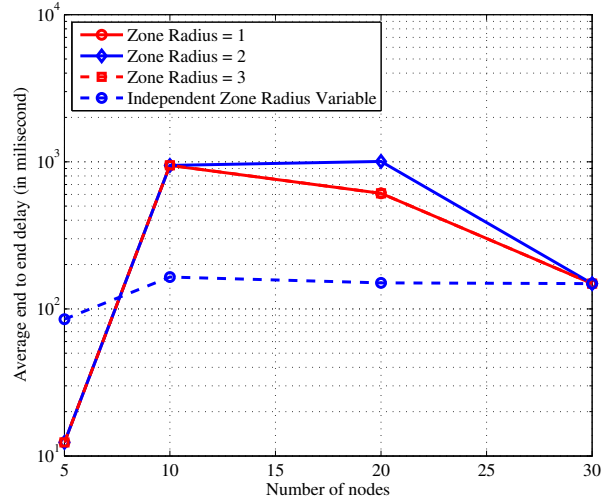


Fig. 7. Comparison results between the average end-to-end delay in ZRP with different zone radii from 1 to 3 and IZRP, as the network size is 5, 10, 20, and 30 nodes, respectively.

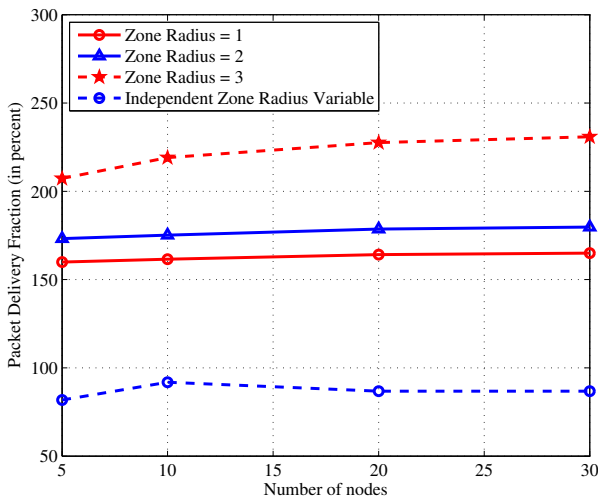


Fig. 6. Comparison results between the packet delivery ratio in ZRP with different zone radii from 1 to 3 and IZRP, as the network size is 5, 10, 20, and 30 nodes, respectively.

could probably be proper for nodes mobility in MANETs, regarding as IZRP, since the fixed zone radii in the regular ZRP could lead to link breakage in the network mobility.

Fig. 6 shows that the number of packets loss in IZRP as different zone radii is much lower than that in ZRP. This is because the nodes in IZR dynamically changes their own zone radii based on the network behavior. This feature minimizes the number of packets loss during the transmission time, since ZR could not resilient its zone radius for balancing routing traffic. Consequently, the reduction results in decreasing significantly the average end to end delay as can be seen in Fig. 7.

V. CONCLUSION AND FUTURE WORK

Due to the nodes mobility in the Mobile Ad Hoc Networks (MANETs), it is hardly to manage the nodes' zone radii for Independent Zone Routing Protocol. Zone radius plays an important role among heterogeneous fashions in MANETs and ability to adapt to the rapidly changing environmental conditions. Therefore, an optimized algorithm has an important role in reduction of control traffic and improvement of throughput in the network mobility.

The paper has been discussed about the efficient zone radius determination algorithm with high efficient performance for nodes mobility in MANETs. Based on the adjacent nodes' radii, a source node can estimate its own optimal zone radius. The simulation results show our first stage which was implemented to the comparison of ZRP and IZRP. As in future work, we are implementing the modified IZR for demonstrating our proposed algorithm in order to prove its efficiency which will be better than the current IZR.

REFERENCES

- [1] P. Samar, M. Pearlman, and Z. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 595 – 608, Aug. 2004.
- [2] H. Wedde, S. Lehnhoff, S. Senge, and A. Lazarescu, "Bee inspired bottom-up self-organization in vehicular traffic management," in *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Sept. 2009, pp. 278 –279.
- [3] M. Pearlman and Z. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395 –1414, Aug. 1999.
- [4] B. Patel and S. Srivastava, "Performance analysis of zone routing protocols in mobile ad hoc networks," in *2010 National Conference on Communications (NCC)*, jan. 2010, pp. 1 –5.
- [5] Z. Haas, M. Pearlman, and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," *IETF, MANET Internet Draft*, Jul. 2002.
- [6] —, "The intrazone routing protocol (iarp) for ad hoc networks," *IETF, MANET Internet Draft*, Jul. 2002.

- [7] D. Ravilla, V. Sumalatha, and P. C. S. Reddy, "Performance comparisons of zrp and izrp routing protocols for ad hoc wireless networks," in *International Conference on Energy, Automation, and Signal*, Dec. 2011, pp. 1 – 8.
- [8] —, "Hybrid routing protocols for ad hoc wireless networks," *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, vol. 2, no. 4, Dec. 2011.
- [9] S. Gandhi, N. Chaubey, P. Shah, and M. Sadhwani, "Performance evaluation of dsr, olsr and zrp protocols in manets," in *International Conference on Computer Communication and Informatics*, Jan. 2012, pp. 1 –5.
- [10] A. Okazaki and A. Frohlich, "Ad-zrp: Ant-based routing algorithm for dynamic wireless sensor networks," in *18th International Conference on Telecommunications*, May 2011, pp. 15 –20.
- [11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep., 2005.
- [12] S. Okdem, D. Karaboga, and C. Ozturk, "An application of wireless sensor network routing based on artificial bee colony algorithm," in *IEEE Congress on Evolutionary Computation*, Jun. 2011, pp. 326 –330.
- [13] T. D. Seeley, *Honeybee Democracy*, Princeton University Press, 2010.
- [14] P. Brijesh, "Zone routing protocol codes, available at the link: <http://magnet.daiict.ac.in/magnetmembers/mtech/2007/patelbrijesh/>," 2009.