Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 431013, 9 pages
http://dx.doi.org/10.1155/2014/431013

Research Article

A Bioinspired Adaptive Congestion-Avoidance Routing for Mobile Ad Hoc Networks

Huang Qiong,[1] Yin Pengfei,[1] Chen Qianbin,[1] Gong Pu,[1] and Yang Xiaolong[2]

[1] *Key Lab of Mobile Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2] *School of Computer and Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China*

Correspondence should be addressed to Huang Qiong; huangqiong@cqupt.edu.cn

Received 13 November 2013; Accepted 6 April 2014; Published 14 May 2014

Academic Editor: Dan Wang

Traditional mobile Ad Hoc network routing protocols are mainly based on the Shortest Path, which possibly results in many congestion nodes that incur routing instability and rerouting. To mitigate the side-efforts, this paper proposed a new bioinspired adaptive routing protocol (ATAR) based on a mathematics biology model ARAS. This paper improved the ARAS by reducing the randomness and by introducing a new routing-decision metric "the next-hop fitness" which was denoted as the congestion level of node and the length of routing path. In the route maintenance, the nodes decide to forward the data to next node according to a threshold value of the fitness. In the recovery phase, the node will adopt random manner to select the neighbor as the next hop by calculation of the improved ARAS. With this route mechanism, the ATAR could adaptively circumvent the congestion nodes and the rerouting action is taken in advance. Theoretical analysis and numerical simulation results show that the ATAR protocol outperforms AODV and MARAS in terms of delivery ratio, ETE delay, and the complexity. In particular, ATAR can efficiently mitigate the congestion.

1. Introduction

A bioinspired mechanism from cell biology called *adaptive response by attractor selection* [1] (ARAS for short) is employed to propose a new adaptive on-demand routing protocol (MARAS for short) [2–4]. ARAS is originally a model for host *Escherichia coli* cells to adapt its behavior such as metabolic synthesis in order to keep the high growth rate in the dynamically changing nutrient condition. The MARAS adaptively select the appropriate next hop according to the solution result of the ARAS mechanism. Comparing with the traditional routing protocol, ad hoc on-demand distance vector routing [5–13] (AODV for short), MARAS provides an effective route recovery to tackle link failures [14–16] without issuing any additional broadcast control message, reduces the average delay, and has higher delivery efficiency.

Unfortunately, the protocols stated above still have many deficiencies [17–19], such that they are mainly based on the condition "Shortest Path," and possibly form many congestion nodes to cause routing instability and rerouting process. AODV and MARAS take the hop count of feedback from the destination as the only parameter to update routing information without considering the impact of heavy load nodes on the routing stability [20, 21]. Furthermore, they also did not take the load balancing into account, which may form local "hotspots" and lead to the selection of an extremely congested node as the next hop. The congested next hop will result in frequent rerouting, trigger more route control packets, aggravate the network congestion, and consume a large amount of network resource.

To overcome these weaknesses, this study also utilizes the adaptability and robustness [22–26] of the bioinspired mechanism ARAS and is concerned about the impact of congestion node while preserving the character of "Shortest Path" in the routing design. In summary, we have three major contributions in this work. Firstly, based on the ARAS, this paper utilizes a cross-layer design method to present a new bioinspired routing-decision metric, that is, the next hop fitness. This route metric could denote the congestion level of node and the length of routing path. Secondly, the original ARAS model in the MARAS utilizes a noise-driven stochastic approach when the routing path deteriorates, and

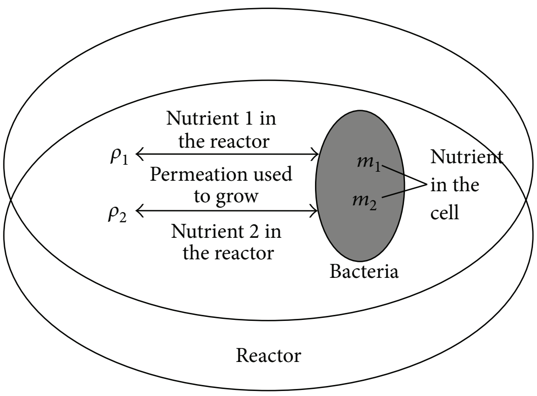


Figure 1: The behavior model of bacteria in the reactor.

its randomicity would lead to a noneffective node selection. This study eliminates its randomicity by redefining the model and the meanings of parameters. The MARAS might suffer a variable and long delay which resulted by the calculation of ARAS at each node. In contrast, we only introduce the improved ARAS into the recovery phase and present a new bioinspired routing recovery algorithm. It can cut down the processing delay and quickly and accurately recover from the failure without using additional broadcast control packets. At last, by using the bioinspired route metric in the routing-decision along with the bioinspired routing recovery, we present a new bioinspired adaptive routing protocol, and it allows the routing-decision node to adaptively select the light-load node as the next hop with shorter path. According to the simulation results, the ATAR protocol has better adaptability and stability; it can efficiently mitigate the congestion and implement the load balancing.

2. Adaptive Response by Attractor Selection

2.1. Attractor Selection-Based Mathematical Model. The ARAS is modeled after the behavior of *E. coli* cells, which is capable of adapting to dynamically changing nutrient conditions without an embedded rule-based mechanism. Figure 1 illustrates the condition when a bacteria cell is in a reactor whose culture media contain two kinds of nutrients necessary for growth of bacteria. The m_1 and m_2 represent the concentrations of nutrient types 1 and 2 in the cell, respectively. ρ_1 and ρ_2 represent the nutrient in the reactor.

According to [1], the equilibrium conditions in the metabolic network are called attractors; the cell volume growth rate of *E. coli* is called activity, α. A mutant *E. coli* cell has a metabolic network consisting of two mutually inhibitory operons which synthesize the two corresponding nutrients 1 and 2. When the environment contains sufficient amount of both nutrients, a cell can live independently in which nutrient is produced. Once one of the nutrients becomes scarce, for example, as the nutrient 1 does, the activity α will decrease; then, the cell autonomously and adaptively begins to control the mRNA (messenger RNA) concentration of the corresponding operon in order to produce the lacking nutrient m_1. The nutrient 1 in the cell will permeate the membrane; then, the nutrient 1 in the reactor would return sufficiently. By this way, the cell regulates the missing nutrient production to make the cell return to a stable equilibrium metabolic condition. By this way, the cell autonomously and adaptively carries "attractor selection" into execution according to the dynamically changing nutrient conditions in the environment. In [4], the dynamics of nutrient concentrations is formulated by the following differential equations:

$$\begin{aligned}\frac{dm_1}{dt} &= \frac{\operatorname{syn}(\alpha)}{1+m_2^2} - \deg(\alpha)\, m_1 + \eta_1,\\ \frac{dm_2}{dt} &= \frac{\operatorname{syn}(\alpha)}{1+m_1^2} - \deg(\alpha)\, m_2 + \eta_2,\end{aligned} \tag{1}$$

where the term η_i corresponds to the white Gaussian noise in the gene network as internal and external noise that affects the nutrient concentrations. The terms syn(α) and deg(α) are the rate coefficient of synthesis and degradation, respectively. The growth rate α changes in accordance with nutrient concentrations in the cell and the reactor.

Leibnitz et al. [2–4] extended the 2-dimensional attractor selection to the M-dimensional case. The extended model can adaptively select one good state from M states according to activity value. The extended model ARAS is expressed as follows:

$$\begin{aligned}\frac{dm_1}{dt} &= f\left(m_1,\dots,m_M\right)\times\alpha+\eta_1,\\ &\vdots\\ \frac{dm_M}{dt} &= f\left(m_1,\dots,m_M\right)\times\alpha+\eta_M.\end{aligned} \tag{2}$$

Note that the resulting vectors have the structure

$$x^{(k)} = [L,\dots,L,H,L,\dots,L], \tag{3}$$

with a single high value H at the kth entry and all others are of low values L.

Figure 2 illustrates the out value of the M-dimensional ARAS with the input of α. From $\alpha = 0$ to 0.3, that is, α is low, therefore, the value of term $f(m_1,\dots,m_M) * \alpha$ will decrease; each value m_i receives more effect from the noise η_i and has a random value. When the maximal solution is found, α starts increasing; the value of term $f(m_1,\dots,m_M) * \alpha$ will increase. Therefore, the gap between selected value and nonselected value grows larger and becomes stable with one high value H and $M-1$ low values L once $\alpha = 1.0$ which indicates that the system reaches the suitable attractor.

Leibnitz successfully adopted the M-dimensional attractor selection in the MARAS protocol for next hop selection among neighbors. In MARAS, M neighbor nodes of routing-decision node are defined as M-dimensional attractors. The term m_i ($1 \le i \le M$), called probability value, indicates whether the ith neighbor should be selected among M neighbors. Activity α ($0 \le \alpha \le 1$) is equivalently regarded as the information which shows the goodness or stability of the current routing condition, and it is updated by the feedback information from the destination.

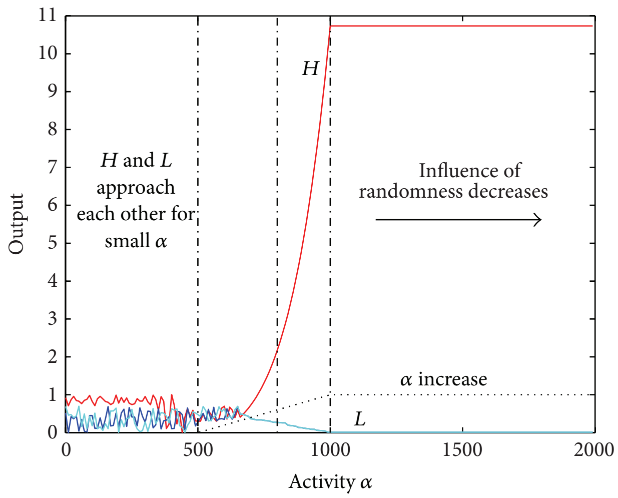


Figure 2: Influence of activity on output in the model [2–4].

Using (2) along with α allows the routing-decision node in MARAS to select the next route node according to the maximum probability value among the M neighbors. When the current routing path becomes unstable or interruptive, the activity decreases. As a result, a larger effect from noise term η_i will take place; the m_i of each neighbor will have a random value. Therefore, the routing-decision node would use a random manner to select the neighbor node with the maximal value. On the other hand, once a suitable neighbor node k is selected, the activity will increase and the effect of noise will be suppressed, which gives a single high value m_k and $M - 1$ low values. This means that only the suitable neighbor will be selected as the next hop, which then allows the routing path to return to a suitable state again or maintain the current routing path for forwarding data.

2.2. Improvement on Randomicity of ARAS Model. In Leibnitz's method, routing-decision node utilizes a noise-driven stochastic approach to choose the next hop when the current routing path deteriorates. Due to the randomicity of the random walk phase, the routing-decision nodes may select a neighbor node which is too far from the destination or heavy load as the next hop especially when there are too many candidate neighbor nodes. Conspicuously, the probability of selecting a noneffective node as the next hop will increase. As a result, an effectual transmission route cannot be swiftly and accurately found, which would result in an increase of the congestion degree of routing node.

To this point, this paper modifies the mathematical model of ARAS as follows: we define the M attractors as the metabolic states of M cells and map the neighbors of routing node to the cells. Each cell has an activity value of its own, that is, growth rate of cell. Let the activity α_i $(0 \le \alpha_i \le 1)$ of the ith cell be the new metric, "next hop fitness," which represents the goodness grade of routing path if the ith neighbor is elected as the next hop. Therefore, each neighbor has a metric to denote the transmission state if it acts as an intermediate node for the destination node. The output values m_i correspond to the packet delivery gain of the current routing-decision node's ith neighbor, which is the improvement degree for packet delivery if the ith neighbor acts as the next hop node.

For all neighbor nodes, the improved M-dimensional model ARAS is described as follows:

$$\begin{gathered}\frac{dm_i}{dt} = \frac{\operatorname{syn}(\alpha_i)}{1 + m_{\max}^2 - m_i^2} - \deg(\alpha_i)\, m_i + (1 - \alpha_i)\,\eta_i, \\ \operatorname{syn}(\alpha_i) = \beta\alpha_i * (\lambda\alpha_i^{\gamma} + \varphi^*), \qquad \deg(\alpha_i) = \alpha_i, \\ \varphi^* = \frac{1}{\sqrt{2}},\end{gathered} \tag{4}$$

where $m_{\max} = \max(m_i)$, $1 \le i \le M$; each differential equation has a stochastic influence from an inherent Gaussian noise term η_i, and it obeys normal distribution with parameters of mean 0.1 and variance 0.01. β and λ are used to restrict the value range of the delivery gain, and γ is used to control the impact of fitness to the delivery gain. In this paper, we get $\beta = 1/10$, $\lambda = 9$, and $\gamma = 2$. The revised ARAS model reduces its randomicity and sufficiently takes advantage of the useful routing information of neighbor nodes.

2.3. Routing Recovery Based on the Improved ARAS. In MARAS, all nodes calculated the probability values of neighbors with the help of ARAS model before making route decision and then selected the next hop according to the maximum. Conspicuously, this method of routing decision would engender large processing delay in the routing process of MARAS. To overcome this drawback, we only introduce the improved ARAS into the local recovery phase and present a new routing recovery algorithm. In the routing decision-making phase, the nodes decide whether to local recovery or forward the date according to the fitness value without compute process by ARAS. Therefore, we can cut down the processing delay. In respect to this mechanism, the route recovery can be achieved without using additional broadcast control packets. The detailed account is as follows.

Firstly, we define the meanings of parameters in the improved ARAS, as expatiated in Section 2.2. We map the M neighbors to the metabolic states of M cells and map the next hop fitness of the ith neighbor to the activity of the ith cell. Secondly, the routing-decision node obtains the delivery gain item and the next hop fitness item of the ith neighbor forms the local neighbor table (as explained in Section 3) and takes them as the initial values of m_i and α_i in (4). If the neighbors never participate in the routing process for the destination, both of m_i and α_i are initialized with zero. Thirdly, the fitness of each possible candidate neighbor in routing path is applied to the ARAS model (4) to compute its packet delivery gain, respectively. Then, the routing-decision node selects the neighbor with the highest gain as the next hop to forward date packet, and we update each delivery gain item in the local neighbor table with the results. Finally, the selected neighbor repeats the above process until it finds out the destination node in the range of local recovery.

2.4. Evaluation of the Routing Recovery Algorithm. In order to evaluate the performance of the improved ARAS model

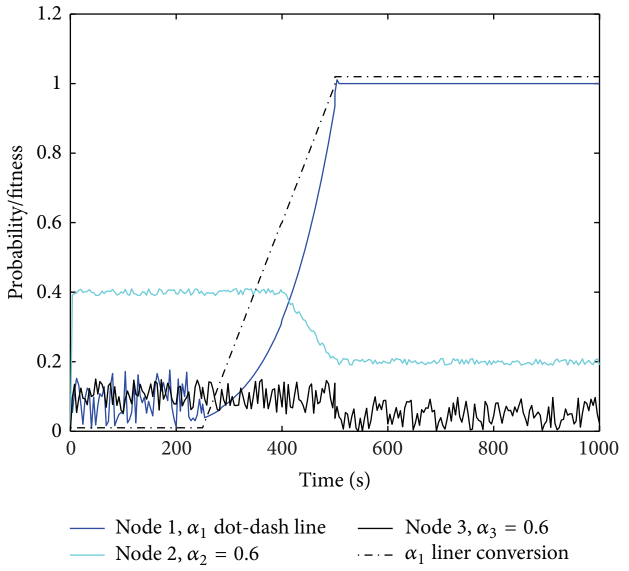


Figure 3: The delivery gain of neighbors on the different values of fitness.

employed in the routing recovery algorithm, this paper assumes that the routing-decision node has three neighbors in the recovery phase. Here, the next hop fitness node 1, that is, α_1, obeys the liner conversion over the time as the dot-dash line shown in Figure 3. The fitness values of nodes 2 and 3 are 0.6 and 0.2, respectively.

According to the condition of fitness, we calculate the delivery gain of all neighbors by (4) with MATLAB. As shown in Figure 3, the output values of m_1 and m_2 fluctuate from 0 to 0.2 due to a stochastic influence from the noise-driven term η_i when $\alpha_1 \leq 0.3$ and $\alpha_3 = 0.2$ (ignoring node 2, $\alpha_2 = 0.6$). When $0.3 < \alpha_1 < 0.7$, the delivery gain values of all nodes are obviously different, and the greater the fitness is, the greater the corresponding delivery gain of neighbor is. When $\alpha_1 \geq 0.7$, the delivery gain values of node 1 and others are in the relation of mutual inhibition, and the gap between them grows larger and becomes stable with maximum m_1. As stated above, if the fitness values are small (less than 0.3), the routing-decision node will adopt random manner to select the neighbor as the next hop, which is as the same manner as the original model in the MARAS. If the fitness values are greater than 0.3, the delivery gain of neighbor is determined by the corresponding fitness, and the routing-decision node selects the next hop according to the maximum. Furthermore, there will be mutual inhibition to maintain the delivery gain of neighbor node at the highest level if the neighbor node has greater fitness than 0.7. Therefore, this study takes the fitness equal to 0.7 as the threshold to decide either to carry out the local recovery or to reinitialize the routing.

The solutions of the multidimensional differential equations vary with the changes of values of the independent variables; such a feature could be utilized in the design of the route behavior. If the fitness of the initial next hop is greater than 0.7, the routing-decision node will keep on selecting the initial next hop in which the corresponding packet delivery gain would be the highest; otherwise, the routing-decision node implements the local recovery. During the local recovery phase, if there are neighbors whose fitness is greater than 0.3, the routing-decision node still can take the neighbor with a higher fitness as the next hop, which takes full advantage of the useful routing information of neighbor nodes. If there are no values greater than 0.3, the routing-decision will utilize the noise-driven random walk. By this approach, we reduce the randomicity of ARAS model to quickly and accurately recover from the failure, especially when there are too many neighbors. The simulation result also shows that the improved ARAS model has better adaptability and stability.

3. A Bioinspired Adaptive MANET Routing to Circumvent Congestion Nodes

In this study, we present a new bioinspired adaptive routing (ATAR) to circumvent the congestion nodes in MANET. The discovery of paths from source to destination is done in the route setup phase by broadcasting route request packets like in the conventional routing methods, for example, AODV. In the process of routing maintenance, the routing-decision nodes update the routing information with a feedback manner the same as that in MARAS and perform the local recovery with the routing recovery algorithm based on the improved ARAS.

Compared with AODV [5], the ATAR protocol adds two items to the route table: next hop fitness item and feedback window item, where feedback window is a list of traveled hop counts of the feedback packet originated at the destination and sent via the current node during a certain time. We also add two items to the local neighbor table: next hop fitness α_i and corresponding packet delivery gain m_i. Besides, we add a new field, "fitness," to the route control message RREP for recording the fitness value of node which sends or forwards the RREP.

3.1. *Account for Fitness.*

The fitness item of route table is a significant parameter for adaption to circumvent congestion nodes with shorter path and decide whether to carry out the local recovery. We calculate the fitness of node by "feedback update" and "decay update"; the α changes according to the two factors in the range from 0 to 1. "Feedback update" is used to keep the routing information up-to-date based on the recent feedback packets which are sent by the destination. "Decay update" is used to avoid utilizing the outdated information. The elaborate descriptions are as the following chapters.

3.1.1. *Feedback Update of Fitness.*

In our protocol, we adopt a cross-layer design method that involves the MAC layer and the network layer to complete the feedback update. The "feedback update" is calculated by two aspects: queue buffer in the MAC, α_{queue} for short, and the traveled hop count of feedback packet, α_{hop} for short. The former one is represented as the congestion degree of node. The latter one is represented as the length and stability of the route path, and it is calculated when the feedback packet arrives. The formula is as follows:

$$\frac{d\alpha_{\text{queue}}}{dt} = \sqrt[3]{\frac{(B-b)}{B}} - \alpha_{\text{queue}}, \quad (5)$$

$$\frac{d\alpha_{\text{hop}}}{dt} = \frac{(w_{\min}+2)}{(w(t)+2)} - \alpha_{\text{hop}}, \quad (6)$$

$$T_{\text{hop}} = 4 * \text{net\_travel\_time}. \quad (7)$$

In (5), B is the total queue buffer of node and b is the size of occupied queue. The greater $(B-b)/B$ is, the greater α_{queue} is. In this study, we define that the node is congested when the queue buffer is occupied up to 65% or more. Therefore, the node is not suitable to act as an intermediate routing node if the vacancy rate of node's queue is less than 35%; that is, $\alpha_{\text{queue}} < 0.7$. In (6), $w_{\min}$ is the minimum traveled hop count in the feedback window during the time T_{hop} before current time. $w(t)$ is the most recent feedback packet's traveled hop count. If the hop count to the destination becomes larger, there will be $(t) > w_{\min}$, and the α_{hop} will decrease; then it means that the current path is unstable and the length is more than before. In (7), the net_travel_time is the longest period for the networks to complete the delivery of a date packet, including queuing delay and propagation delay. Consider the following:

$$\alpha = \begin{cases} \alpha_{\text{hop}}, & \text{if the available ratio of queue} \geq 35\% \\ \alpha_{\text{queue}}, & \text{otherwise.} \end{cases} \quad (8)$$

In this study, we give priority to consider the node's load condition; meanwhile, we select the routing path as short as possible. As a result, we could reduce the extra delay caused by the longer routing path which is due to the fact that the date packets steer clear of the local "hotspots" or circumvent the congestion nodes. Therefore, the node's fitness α, which is defined for the specified destination node, should be calculated by α_{queue} and α_{hop}; the format is expressed as (8).

3.1.2. Decay Update of Fitness. At each intermediate node, by using the information from the feedback packet, the fitness is calculated and the routing information is updated. As the feedback packets are sent only when the RREQ or data packets arrive at the destination, it can be concluded that if no packet arrives at the destination, then the intermediate node's fitness will never be carried out by the process of "feedback update." Moreover, we can assume that the intermediate is no longer suitable as the next hop for the route node when the intermediate node is not used for a long time. In order to recover from such situations, the previous learned values of fitness need to be constantly decayed over time to avoid using the outdated information. Therefore, the fitness on each node and route table must be decayed over time before the node participates in the routing process. The formula is as follows:

$$\alpha_{\text{new}} = \alpha_{\text{strored}} - \frac{T_{\text{now}} - T_{\text{last}}}{T_{\text{decay}}} * \eta, \quad (9)$$

$$T_{\text{decay}} = 2 * w(t) * \text{node\_travel\_time}, \quad (10)$$

where the decay constant $\eta = 0.1$, T_{decay} is the basic interval to periodically perform decay process, T_{last} is the latest time to perform decay process, T_{now} is the current time, and node_travel_time is the average period for one node to deliver a date packet to its adjacent node. The further the distance from the intermediate node to the destination is, the longer the period for intermediate node to receive the feedback packet is. Hence, the intermediate nodes, which are far from the destination, will decay too fast due to the small basic interval and result in the failure of the whole routing path. Therefore, the basic decay interval should increase along with the increase of the distance from the destination. So the T_{decay} should be changed with $w(t)$ by (10). The fitness decay mechanism is performed with (9) as long as the nodes use the information of fitness, and the fitness decay mechanism should be suspended if α_{strored} equals zero.

3.2. An Adaptive Routing Protocol to Circumvent Congestion Nodes with ARAS (ATAR)

3.2.1. Route Establishment. We adopt the broadcasting route discovery mechanism from AODV to establish the route. The process is described as follows in detail. Firstly, the source node broadcasts the RREQ with a unique ID to its neighbor nodes. The intermediate node gets the nonreduplicate RREQ and then establishes the reverse path. If the α_{queue} of current intermediate node is less than 0.7, this node discards the RREQ as it is too congested to act as the best candidate of the next hop node. Otherwise, this node would rebroadcast the RREQ. The other intermediate nodes repeat the process until the RREQ reaches the destination. Secondly, the reverse path for the RREP is memorized when the RREQ arrives at the destination. The α_{hop} of destination node is initialized with 1, and the α_{queue} is calculated with its own queue buffer information. Thus, we get the fitness of destination by (8). Then, the RREP, which contains the fitness of destination node, is generated and forwarded in a unicast manner to the source via the reverse path. Thirdly, when the RREP arrives at the intermediate node in the reverse path, the intermediate node sets up the obverse route entry in the route table and the next hop fitness item in the neighbor table for the previous hop with the fitness value which is from the RREP; meanwhile, it adds the traveled hop of RREP to the feedback window item and initializes α_{hop} with 1 and the corresponding delivery gain m_i of the previous hop with 1 (all other neighbors are 0). Then, this node calculates its α and updates the fitness item of RREP. Afterward, this node forwards the renewed RREP again via the reverse path. The other intermediate nodes repeat the process until the RREP reaches the source. Finally, the source node initializes the

Table 1: The complexity of ATAR, MARAS, and AODV.

Items		AODV	MARAS		ATAR	
Time complexity	Initialization	$O(2d)$	$O(2d)$		$O(2d)$	
	Failure occurs	$O(2d)$	Complete the recovery process in the recovery range			
			Yes	No	Yes	No
			$O(2d)$	$O(4d)$	$O(2x)$	$O(2x + 2d)$
Communication complexity	Initialization	$O(2N)$	$O(2N)$		$O(2N)$	
	Failure occurs	$O(2N)$	Complete the recovery process in the recovery range			
			Yes	No	Yes	No
			$O(j + z)$	$O(j + z + 2N)$	$O(i + y)$	$O(i + y + 2N)$

N: the number of nodes in the network; d: network diameter; x: the recovery range TTL of failure node; y: the number of nodes which participate in the recovery process in the range of TTL.
z: the number of nodes which participate in the recovery process in the range of network diameter.
i: the number of neighbors of nodes in y; j: the number of neighbors of nodes in z.

route in the same manner at all the intermediate nodes; then, the date packet forwarding begins.

3.2.2. Route Maintenance. After the route is set up, the data packets are forwarded to the destination by using the information stored in the route entry. The RREP would be sent back to the source to learn the current condition for route maintenance. The detail is as follows. Firstly, the routing-decision node carries out the decay update of fitness for the fitness value in the route table. If the α in the route table of routing-decision node is less than 0.7, the previous next hop in the route table would be too congested or the route path becomes too long; consequently, the routing-decision node should carry out the local routing recovery algorithm. Otherwise, the current node should forward the date packet, and the next hop updates the reverse path and lifetime when it receives the date packet. Then, those particular nodes in the routing path repeat the process until the date packet arrives at the destination. Secondly, the destination node would send back the RREP with its fitness once the date arrives at it. Thirdly, once the particular intermediate node receives the RREP, it updates the corresponding obverse route table entry and neighbor table entry for the previous hop with the value obtained from the RREP as the same manner in the route establishment phase. Finally, the whole route path has been updating.

With the update of route table in accordance with the information of feedback packet, the ATAR could adaptively circumvent the congestion nodes whose queue buffer is occupied up to 65% and more, and the rerouting action is taken in advance. In the recovery phase, the failure nodes could quickly and accurately recover from the failure without using additional broadcast control packets. Therefore, the route failure due to congestion could be avoided, the date packet drops and queue delay reduces, and the successful delivery ratio could be improved.

As shown in Table 1, we compare the ATAR, MARAS, and AODV in terms of time and communication complexity for observing the respective resource consumption. The time complexity is defined as the steps needed to run during the phase of routing, and the communication complexity is defined as the number of messages required to send during the phase of protocol operation. Those protocols all adopt a manner of broadcast in the route initialization; the route control messages cross the network twice and should be distributed process in each node. As a result, the time and communication complexity are $O(2d)$ and $O(2N)$ in the route initialization stage, respectively. In the failure scenarios, the complexity of those protocols will be different in that they carry out different route recovery schemes. Table 1 shows that generally the ATAR protocol does not consume too much resource, and it acquires high efficiency and overcomes the drawbacks of heavy overhead in the MARAS.

4. Evaluation

4.1. Simulation Settings. We evaluated the performance of the proposed protocol using OPNET. We divided a flat area into two equal flat areas and placed three fixed nodes at the regional center, and the other nodes were randomly distributed and moved in each area. Two pairs of nodes sent and received the date packets; the source and the destination were in different areas, respectively. The data packets sent by the source in one area would pass through the fixed nodes in the center to reach the destination in the other area.

4.2. Simulation Results. In Figure 4, we show the effects of the number of nodes M when the speed v is 10 m/s. The more nodes in the network, the bigger the probability of end-to-end links would be established is, and the higher the probability that packet will be successfully delivered to the destination. Simulation results show that the ATAR is less susceptible to this effect than the AODV and MARAS. In Figure 5, we show the effects of the speed v when the number of nodes is 37. Simulation results demonstrate that no matter how the node speed changes, the ATAR always performs better than the AODV and MARAS; it means that the ATAR has better adaptability and robustness against the dynamic changing topology. In Figures 6 and 7, we set $M = 37$, $v = 10$ m/s. We collect the number of data packets forwarded by the three fixed nodes in the center in Figure 6. Simulation results show that the load balancing degree of ATAR is obviously better than that of AODV and MARAS. Due to the fact that latency time of data packet in the congestion node buffer

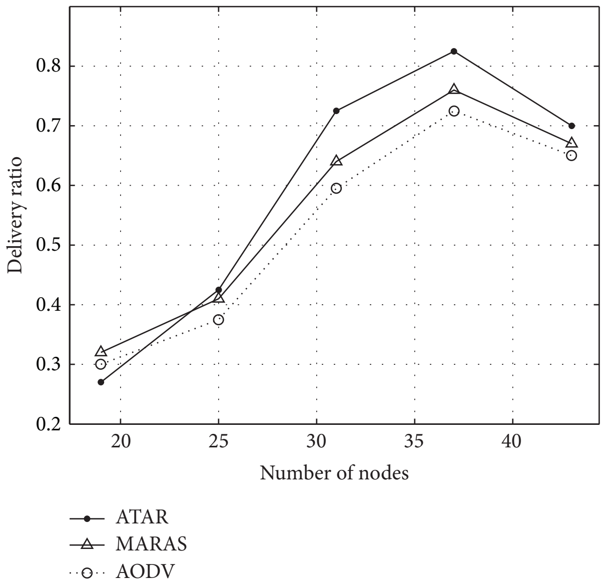


Figure 4: The delivery ratio versus number of nodes.

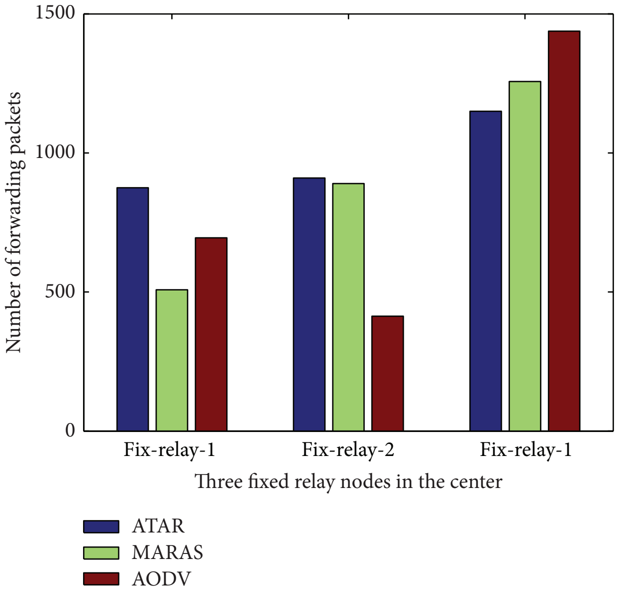


Figure 6: The number of packets forwarded by the fixed nodes.

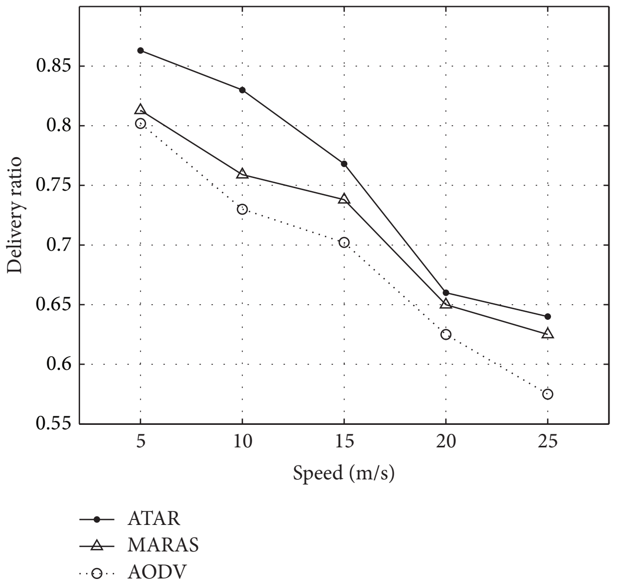


Figure 5: The delivery ratio versus the node speed.

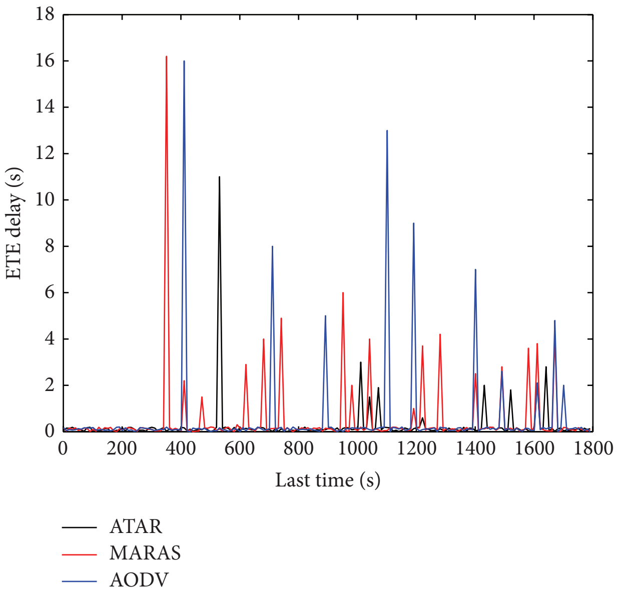


Figure 7: ETE delay versus last time.

would become longer, the ETE delay of a pair of source and destination would increase rapidly to form the spike delay, as the "pulse" shown in Figure 7. Figure 7 shows that the probability of spike delay taking place in ATAR is smaller than that of AODV and MARAS, and the amplitude of pulse is also smaller. The simulation results illuminate that the ATAR is adaptive to choose an idle node as the next hop to circumvent the "hotspot" zone and reduce the latency time due to congestion. In Figure 8, we set the packet size as 256, 512, 1024, 2048, and 4096 (bits), respectively. When there is too heavy load in the network, the average delay is degraded due to congestion node and link competition. The average delay of ATAR changes slower than that of AODV with an increasing size of packet, and we can see that the ATAR is less susceptible to the negative effect of heavy load than AODV.

All the simulation results and analyses verify that the ATAR outperforms the AODV and MARAS. Specifically, the number of nodes and load level can significantly affect the network performance, and the proposed scheme ATAR can efficiently control the congestion and share the load.

5. Conclusions

In this paper, we proposed a new bioinspired adaptive routing protocol (ATAR). In the routing initialization and

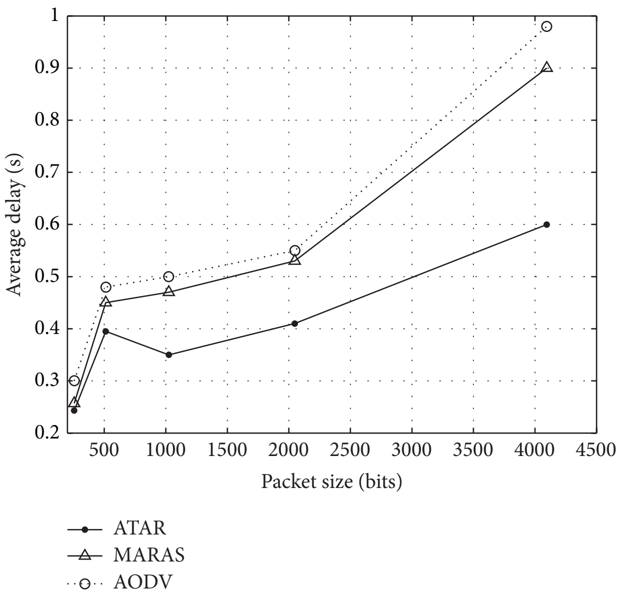


FIGURE 8: Average delay versus packet size.

routing maintenance, the intermediate nodes forward a route request and implement route recovery according to the fitness of routing table. In the local recovery, we employ the proposed routing recovery algorithm to calculate the packet delivery gain of each possible candidate neighbor in the route path, and the routing-decision node chooses the one with highest gain among these neighbor nodes as the next hop. Simulation results show that ATAR achieves better performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by 973 Program (no. 2012CB3 15803), the National Natural Science Foundation of China (no. 60972070), the Program for Changjiang Scholars and Innovative Research Team in University (IRT1299), and the special fund of Chongqing key laboratory (CSTC).

References

[1] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, "Adaptive response of a gene network to environmental changes by fitness-induced attractor selection," *PLoS ONE*, vol. 1, no. 1, article e49, 2006.

[2] K. Leibnitz, N. Wakamiya, and M. Murata, "Self-adaptive ad-hoc/sensor network routing with attractor-selection," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, San Francisco, Calif, USA, November 2006.

[3] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata, "Extension and evaluation of biologically- inspired routing protocol for manets," IEICE Technical Report, 2009.

[4] K. Leibnitz, N. Wakamiya, and M. Murata, "A bio-inspired robust routing protocol for mobile ad hoc networks," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 321–326, Honolulu, Hawaii, USA, August 2007.

[5] C. E. Perkins, E. M. Belding-Royer, and S. Das, *Ad Hoc on-Demand Distance Vector (AODV) Routing*, RFC 3561, 2003.

[6] S. R. Birdar, K. Majumder, S. K. Sarkar, and C. Puttamadappa, "Performance evaluation and comparison of AODV and AOMDV," *International Journal on Computer Science and Engineering*, vol. 2, no. 2, pp. 373–377, 2010.

[7] W. Ge and P. Li, "An optimized AODV protocol for Ad Hoc network," in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, vol. 2, pp. 1–4, October 2008.

[8] M. K. Mishra, "A trustful routing protocol for adhoc network," *Global Journal of Computer Science and Technology*, vol. 11, no. 8, 2011.

[9] Z. Heying, L. Baohong, and D. Wenhuan, "A rate and queue controlled active queue management," *Acta Electronica Sinica*, vol. 11, no. 31, pp. 1743–1746, 2003.

[10] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.

[11] C. Adjih, E. Baccelli, T. H. Clausen, P. Jacquet, and G. Rodolakis, "Fish eye OLSR scaling properties," *Journal of Communications and Networks*, vol. 6, no. 4, pp. 343–351, 2004.

[12] J. Chen, Y. Z. Lee, D. Maniezzo, and M. Gerla, "Performance comparison of AODV and OFLSR in wireless mesh networks," in *Proceedings of the IFIP 5th Annual Med- iterranean Ad Hoc Networking Workshop (Med-Hoc-Net '06)*, pp. 271–278, June 2006.

[13] L.-L. Zhang and C.-S. Li, "Comparison analysis for MANET routing protocol," *Acta Electronica Sinica*, vol. 28, no. 11, pp. 88–92, 2000.

[14] F. D. Rango, J.-C. Cano, M. Fotino, C. Calafate, P. Manzoni, and S. Marano, "OLSR vs DSR: a comparative analysis of proactive and reactive mechanisms from an energetic point of view in wireless ad hoc networks," *Computer Communications*, vol. 31, no. 16, pp. 3843–3854, 2008.

[15] J. Wang and S. Lee, "A performance comparison of swarm intelligence inspired routing algorithms for MANETs," in *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA '09)*, pp. 432–442, 2009.

[16] H. F. Wedde, M. Farooq, T. Pannenbaecker et al., "BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 153–160, June 2005.

[17] A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8–27, 2002.

[18] M. Meisel, V. Pappas, and L. Zhang, "A taxonomy of biologically inspired research in computer networking," *Computer Networks*, vol. 54, no. 6, pp. 901–916, 2010.

[19] S. Ahmed and M. S. Alam, "Performance evaluation of important ad hoc network protocols," *Eurasip Journal on Wireless Communications and Networking*, vol. 2006, Article ID 078645, 2006.

[20] E. Borgia and F. Delmastro, “Effects of unstable links on AODV performance in real testbeds,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2007, Article ID 019375, 2007.

[21] A. Gupta, Y. Chauhan, S. Sharma, and S. Bhardwaj, *Performance Evaluation of Mobile Ad Hoc Network Under Varied Traffic*, NCEC, Abhipur, 2010.

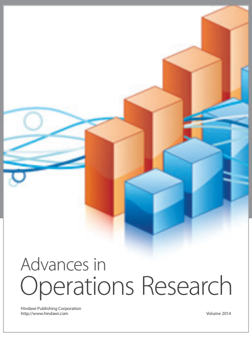
[22] K. Leibnitz, N. Wakamiya, and M. Murata, “Resilient multipath routing based on a biological attractor selection scheme,” in *Proceedings of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (BioAdit ’06)*, Osaka, Japan, January 2006.

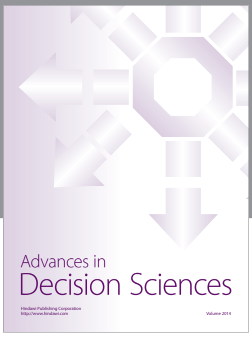
[23] F. Dressler and O. B. Akan, “A survey on bio-inspired networking,” *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.

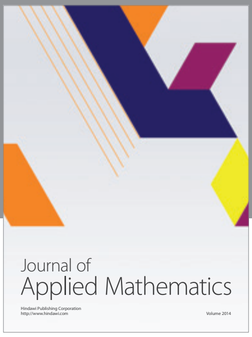
[24] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata, “Extension and evaluation of biologically-inspired routing protocol for manets,” IEICE Technical Report, 2009.

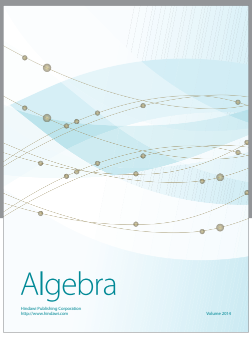
[25] F. Ducatelle, *Adaptive routing in ad hoc wireless multi-hop networks [Ph.D. thesis]*, University della Svizzera Italiana, 2007.

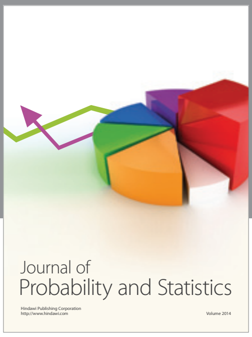
[26] K. Leibnitz, N. Wakamiya, and M. Murata, “Biologically inspired self-adaptive multi-path routing in overlay networks,” *Communications of the ACM*, vol. 49, no. 3, pp. 63–67, 2006.


Advances in
Operations Research

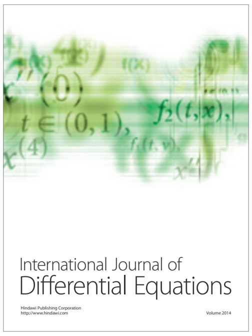

Advances in
Decision Sciences


Journal of
Applied Mathematics

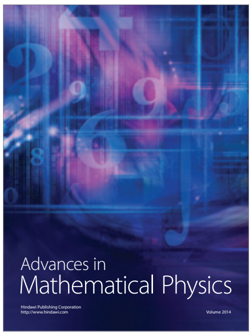

Algebra

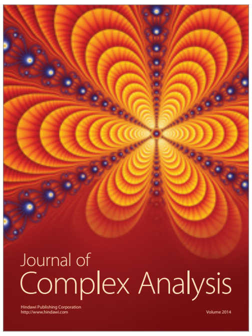

Journal of
Probability and Statistics

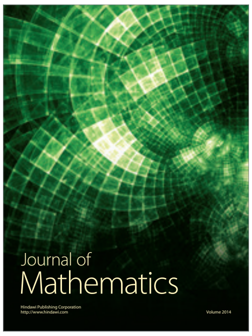
The Scientific
World Journal

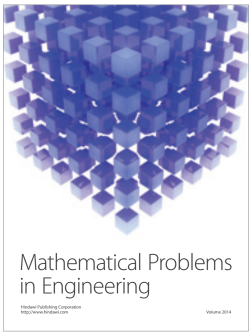

International Journal of
Differential Equations

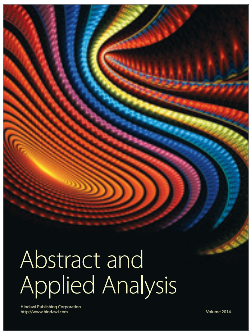
International Journal of
Combinatorics

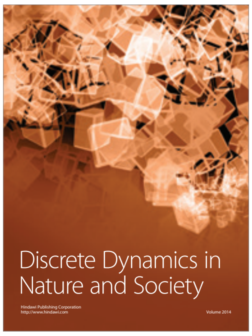

Advances in
Mathematical Physics

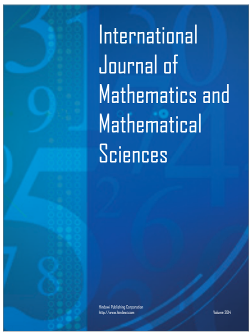

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

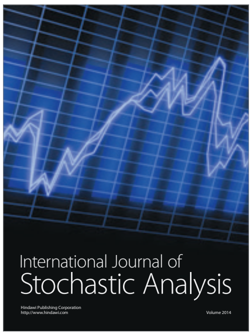

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization