Performance Comparison of the Neuron Routing Algorithm for Mobile Ad Hoc Networks

V.E Mujica-V, Y. Rebahi, D. Sisalem, R. Popescu-Zeletin
Fraunhofer FOKUS Institute, Berlin,Germany
{mujica,rebahi,sisalem,zeletin}@fokus.fraunhofer.de

V.E. Mujica-V
Fraunhofer FOKUS Institute
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
Telf.: +49 30 3463 9488
Fax: +49 30 3463 8000
Email: mujica@fokus.fraunhofer.de

Abstract
This paper proposes a strong research interest in a field related to bio-informatics and computational biology in Mobile Ad Hoc Networks (MANET). Basically, we develop an intelligent mobile agent called the NEURAL Agent, which improves the performance of the NEUron Routing ALgorithm (NEURAL). This autonomous agent coordinates the gathering of information by integrating, planning, scheduling, and making-decision procedures with other agents through different modules that operate asynchronously. In addition, a comparison study is carried out based on the performance of the Dynamic Source Routing (DSR), and the Ad-Hoc On-Demand Distance Vector Routing (AODV).

Keywords
Routing, autonomous agents, MANET, NEURAL, DSR.

Performance Comparison of the Neuron Routing Algorithm for Mobile Ad Hoc Networks

Vicente E. Mujica-V., Yacine Rebahi, Dorgham Sisalem and Radu Popescu-Zeletin
Fraunhofer FOKUS Institute. Berlin, Germany.
{mujica,rebahi,sisalem,zeletin}@fokus.fhg.de

***Abstract*— This paper proposes a strong research interest in a field related to bio-informatics and computational biology in Mobile Ad Hoc Networks (MANET). Basically, we develop an intelligent mobile agent called the NEURAL Agent, which improves the performance of the NEUron Routing ALgorithm (NEURAL). This autonomous agent coordinates the gathering of information by integrating, planning, scheduling, and making-decision procedures with other agents through different modules that operate asynchronously. In addition, a comparison study is carried out based on the performance of the Dynamic Source Routing (DSR), and the Ad-Hoc On-Demand Distance Vector Routing (AODV).**

I. Introduction

The design of routing protocols has recently been expanding in the MANET research areas [1], [2], [3], [4], [5]. *Routing* provides the methods and rules for discovering paths along which information can be sent over the network. To develop routing protocols, the networking architectures can be characterized with respect to the following properties: **self-organizing**; reacting locally to environment and context changes, **autonomously controlled**; applying *local* control optimization strategies, and **distributed**. In a prior work [1], the above features was achieved through the application of stochastic and probabilistic search techniques in oder to develop a novel routing algorithm for MANET called the NEURon Routing ALgorithm (NEURAL). Stochastic search algorithms strongly employ randomized decision-making procedures to seeking for alternative solutions to a given problem. NEURAL integrates the learning and self-organizing abilities of the brain into the area of MANETs. More exactly, the protocol is inspired by the synapses process between neurons, when a signal is propagated. Basically, the most significant characteristic of NEURAL is the uniform distribution of the information around the node's location, based on the current changes in its environment. Using a 2-hop acknowledgment mechanism, local information is monitored to perform route selection method, classification procedures and learning algorithms.

In this paper, we propose the development of a NEURAL Agent through a cross-layer design. The NEURAL Agent coordinates the gathering of information by integrating, planning, scheduling, making-decision procedures, and coordination with other NEURAL agents through different modules that operate asynchronously.

This paper is organized as follows: Section II introduces terms and concepts in the area of neurobiology and Artificial Intelligence, as well as, their respective interrelationships within the context of the MANETs. The section III describes the system components, modules, and functions for the NEURAL Agent. A simulation work validating the improvements of NEURAL Agent is discussed in section IV. And finally, we offer conclusions and a short sketch of further works in section V.

II. Associations between Neurobiology and MANET

In this section, a brief description of the major components and terms in neuro-science is shown in ordinary (unitalicized) print, whereas the respective relationship of this terms is emphasized using italics in the context of MANETs.

Neuron: A neuron is the nerve cell of the nervous system, which uses biochemical reactions to receive, process and transmit a *nerve* impulse rapidly through the nervous system. *In MANETs, a neuron represents an individual agent which computes information from the environment by interacting with other agents. In the rest of the paper, the term node can also denote an agent (i.e. The NEURAL Agent).*

Synapse: The synapse forms the circuits in which the neurons of the central nervous system interconnect. They also provide the means through which the nervous system connects to and controls the other systems of the body. The synapse involves a conjunction of processes that describe the rules whereby the neurons propagate a *nerve* impulse. *In MANETs, **Routing** forms logical connections along which the packets are sent. Thus, routing protocols provide the rules that an agent or node follows to calculate routes towards the destination.* This interrelationship between **Synapse** and ***Routing*** is the main contribution to promoting research in the area of MANET involving Artificial Intelligence. More precisely, it was the inspiration to design the NEUron Routing ALgorithm [1].

Signaling across chemical synapses: The arrival of a nerve impulse produces an **activation potential** which represents as internal behavior of a neuron characterized by secreting specific chemicals “excitatory” and “inhibitory” molecules. A neuron forms a **excitatory** synapse when the activation potential achieves a certain threshold, so that the first neuron fires the second one propagating the signal. Otherwise, the **inhibitory** synapse appears when the intensity of the nerve impulse is not enough to reach this threshold. In consequence, the first neuron inhibits the second one from firing (Blocking of the propagation). *In NEURAL, an agent alternates between* *the two activation potential stages according to its current configuration parameters. An* ***excitatory stage*** *is represented by a group of "optimal" properties which characterizes an agent as favorable to execute some specific actions. For example in the routing area, an intermediate node, which contains the shortest path towards the destination or highest battery level, presents a high probability of being selected as next hop to forward the packet. Otherwise, an* ***inhibitory stage*** *is characterized by agents, which present some weakness. For example, low battery, lack of routing information, etc. These two activation potential stages are explained in detail in section III*

III. THE NEURAL AGENT

In the following, we address the issue of cross-layer networking with the purpose of structuring the NEURAL Agent. In the **Routing Module**, the *Routing Algorithm* is responsible for seeking and computing the path towards the destination. The *Synchronization/Replication* module constantly updates the routing information using interactions with the lower layers (Link Layer). And finally, the *Policy* module analyzes the received routing information for communication to upper layer (i.e. Application Layer).

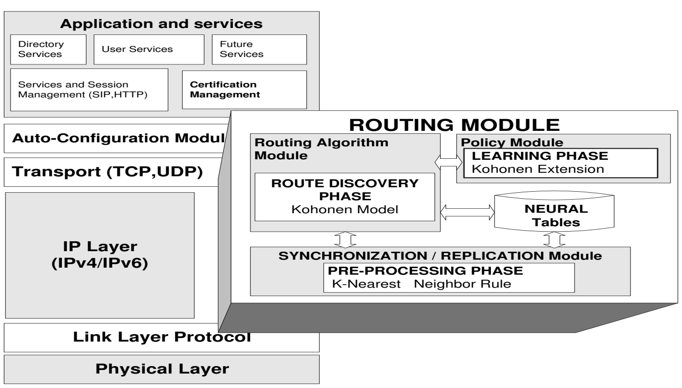

Fig. 1. Cross-layer design for the NEURAL Agent

A. The Pre-processing Phase in the Synchronization/Replication module

The *pre-processing* phase in the NEURAL Agent consists of transformation procedure that involves a classification algorithm. The K-Nearest-Neighbors Rule (K-NNR) [6] is proposed to adapting local variables from the neighborhood into an *activation potential* stage ($A_{\rm s}$) for the NEURAL agent. This stage represents an estimation of the local density based on the 1-hop distance information and it presents two behaviors:

Excitatory: NEURAL Agents belonging to an excitatory class are in crowded areas with a high probability of interconnection or form circuits (synapses). Nodes with an excitatory behavior are identified by the activation stage ($A_{\rm s}$) equal to +1.

Inhibitory: Otherwise, the inhibitory class is identified by isolated regions in the ad hoc network product of the dynamical topology changes. This means that those NEURAL Agents present a low communication capability with their neighbors. The activation stage ($A_{\rm s}$) is set to -1.

Each NEURAL Agent contains a pre-processing interval time ($\tau_{\rm PP}$) after which the preprocessing phase is constantly scheduled. For instance, a node n_X broadcasts a beacon packet in the neighborhood (within the 1 hop distance). Afterwards, K neighbors respond with their *activation potential* stage (A_s), where K_A nodes were found with an excitatory behavior ($K_A < K$). Lastly, the probability that the node $n_{\rm X}$ belongs to an excitatory class is computed using the K-NNR equation as

$$p(A_{\rm s} = +1|n_{\rm X}) = \frac{K_{\rm A}\ \epsilon\ (A_{\rm s} = +1)}{K} \quad (1)$$

The node $n_{\rm X}$ calculates the probability of presenting an inhibitory behavior as $p(A_{\rm s} = -1|n_{\rm X}) = 1 - p(A_{\rm s} = +1|n_{\rm X})$. Finally the current activation potential stage ($A_{\rm s}^{curr}$) is assigned according to the highest probability value as

$$p(A_s^{curr}) = max(\ p(A_{\rm s} = +1|n_{\rm X})\ ,\ p(A_{\rm s} = -1|n_{\rm X})\ ) \quad (2)$$

The Replication/Synchronization module improves the performance of the pre-processing phase in NEURAL due to the fact that this mechanism was adapted to sensing the most recent information from the lower layer (e.g Link layer). More precisely, the continuous changes in the topology network are monitored by setting the *activation potential* stage in the NEURAL Agent based on a measurement of the density around a determined NEURAL Agent.

B. The Route Discovery Phase as Routing Algorithm Module

Similar to the Extremely Opportunistic Routing (ExOR) protocol [5], the NEURAL Agent determines the path as the packet moves through the network. To choose the efficient way to the destination, NEURAL applies in the discovery phase a *competitive learning* model called Self-Organizing Kohonen Model (KSOM) [7], which is employed for self-organized feature maps in neural networks. KSOM describes the **synapse** as a *competition* between neurons.

The Route Discovery phase is configurated by the NEURAL Agent based on the information within a **local activation zone**. For instance, a NEURAL Agent ***i*** is seeking a new route within a group of nodes that are in a *local activation zone*. This zone takes into account the diameter of a 2-hop distance around the agent *i*, which contains "first" and "second" neighbors. The so-called **"first" neighbors** are the nodes within the range of the agent *i*. Similarly, agents in the surrounding of the "first" neighbors are the **"second" neighbors**. To select the next node to forwarding the packet, the agent *i* needs the **Round Trip Time** (RTT) and the number of second neighbors in its *local activation* zone. The **Round Trip Time (RTT)** is a measure of the time it takes to send a packet from agent *i* to a first neighbor agent *y*, and receive a response back at *i*. Following, the mathematical formulation is explained in unitalicized font and then its respective implementation for MANET is clarified in italics. The KSOM is carried out in three processes as

1) Broadcasting of the input: Suppose an incoming stimulus in the neuron *i*, which decides to forward a stimulus towards one of its *m* neighbor neurons. An incoming signal pattern is denoted by the input vector **x** ($\mathbf{x} = x_1, x_2, ..., x_{\mathrm{m}}$). In addition, the synapse of a neural network provides a memory parameter which is associated with the synaptic weight vector **w**. For example, the synapse weight vector between the central neuron ***i*** and its *m* neighbor neurons is denoted by $\mathbf{w}_{\mathrm{i,j}} = w_{\mathrm{i,1}}\ ,\ w_{\mathrm{i,2}}, ..., w_{\mathrm{i,m}}$. Thus, the neuron *i* forms a weight sum as

$$net_{\mathrm{i}} = \sum_{\mathrm{j=1}}^{m} w_{\mathrm{i,j}} \cdot x_{\mathrm{j}} \tag{3}$$

In our approach, the so-called NEURAL agent ***i****, which represents a intermediate node searching a next hop to configure the path to the destination, executes a broadcast action as follows. A* **FIRST_STIMULUS** *packet is broadcasted to the "first" neighbors with a ID number (StimulusID), timestamp (TS), destination (DST) and source (SRC) addresses. The "first" neighbors then check if the DST address has been stored before in one of the neighbors' routing table. A notification (***ROUTE_REPLY***) is sent back to the agent* ***i*** *when the DST address exists. Otherwise, a* **SECOND_STIMULUS** *packet is forwarded to the second neighbors (with the same StimulusID of the FIRST_STIMULUS packet). The SECOND_STIMULUS packet senses the number of second neighbors for each first neighbor agent. An answer (***SECOND_REPLY***) is transmitted back to first neighbors carrying the activation potential stage (A_s) of these second neighbor agents, and their timestamp. Based on the arrival of SECOND_REPLY packets to first neighbors, a second respond (***FIRST_REPLY***) is created using the previous information (nodes 2-hop away from* ***i****) as the number including the second neighbors (N_m), the timestamp (TS) and the activation potential stage. The activation potential stage represents the density of the region which are 2-hop away from the node* ***i*** *and TS is used by the NEURAL agent* i *to calculate the RTT for each first neighbor. Finally, the agent* ***i*** *can configure the* ***x*** *and* ***w*** *vectors as shown equations 4 and 5.*

$$x_{\mathbf{j}} = \exp^{-(\frac{RTT_{\mathrm{j}}}{\tau_{\mathrm{x}}})}, \quad j = 1, 2...m \tag{4}$$

$$w_{\mathbf{i,j}} = \frac{N_{\mathrm{j}}}{\tau_{\mathrm{w}}}, \quad j = 1, 2...m \tag{5}$$

Where RTT_j is the Round Trip Time between the agent ***i*** *and its* ***m*** *"first" neighbors. This value is stored in the input vector* ***x****. The number of second neighbors detected by each "first" neighbor is represented by N_j ($j = 1, 2...m$). Both vectors (***x*** *and* ***w****) are normalized with τ_x and τ_w, which are the maximum values of the RTT and N_j observed in the local activation zone of the node* ***i****. The RTT was selected as the input vector* ***x*** *because it is a measurement of distance between the central point (agent* i*) and its neighborhood (only the first neighbors). Thus, this distance is an important parameter for routing issue. In neural networks, the synaptic weight vector is a parameter which varies accordingly when the neuron's location changes. In the same way, the movement of a particular agent (i.e. the node* i*) in MANET produces that density around its neighborhood constantly changes. Thus, we associated the number of second neighbors as an probabilistic weight which constantly varies such as the synaptic weight vector* ***w*** *in a neural network.*

2) Selection of the "winner": In the KSOM, the above step describes an active neuron ***i***, in which the total excitation is concentrated within a single and connected *cluster* of ***m*** consecutive neurons. Kohonen suggests an approximation (net_{i}) for the eqn. 3 using the position of the maximum excitation. The eqn. 6 sums up the essence of the competition process among the neurons, where a particular neuron *b* satisfies this conditions and it is called the *wining neuron* for the input vector **x** and **w**. To calculate the second term of the Eqn. 6, Kohonen employed in [7] the use of the maximal correlation between two sets of nonlinear related variables. The main idea of this approach was to measure the position within the group of neighbors neurons, where the maximum excitation signal appeared.

$$net_{\mathrm{i}} = \sum_{\mathrm{j=1}}^{m} w_{\mathrm{i,j}} \cdot x_{\mathrm{j}} = \mathbf{max}\ (\sum_{\mathrm{j=1}}^{m} w_{\mathrm{i,j}} \cdot x_{\mathrm{j}}) \tag{6}$$

To implement the so-called "Selection of the winner" in NEURAL, the ***winner*** *agent denotes the "first" neighbor agent with the "optimal properties" to forward a packet to the destination. The "first" neighbors (nodes located 1-hop away) initiate a competition to be selected as "winner". In our case, suppose that the node* ***i*** *has m "first" neighbors. The best-matching agent (winner) is selected by the position where the "maximum excitation signal" occurs. More precisely, after the "broadcasting" action, the node* ***i*** *selects the neighbor agent which presents the maximum value in the sum vector net_i from the eqn. 6. Afterwards, the packet is forwarded to the winner, and the propagation of the "data" packet continues towards the destination. This new entry is stored in the routing table of the NEURAL agent for following packets containing the same destination address.*

3) Adaptation procedure: Ritter et al. [8] developed an adaptation step for the Kohonen model, in which every change in the synapse weight vector (**w**) is limited to a neighborhood zone about the excitation center (neuron *i*). In this zone, these changes that are a product of a subsequent re-occurrence of the same stimulus will lead to an increased excitation in the neighborhood (**learning**). For the NEURAL Agent, this step is explained in the following section.

C. A Learning Phase in the Policy Module

The Policy module analyzes the received information in the routing module to provide a strategy that enforces packet forwarding among nodes based on *reputation* rates. Sensing local *reputation* allows calculation of a control output variable, that is described in the extended model of Kohonen [8]. In a prior work, a *reputation* mechanism was developed in [9] which performs using the Dynamic Source Routing

(DSR) algorithm [10]. In this section, the aim is to integrate this *reputation* mechanism in the proposed NEURAL Agent. The mentioned *reputation* mechanism provides a distributed and efficient monitoring tool for both resource selection and learning issues. In particular, the reputation rates are computed based on three layers mechanism. The lowest layer refers to the Encounter Management (EM). "Encounter" events are represented in NEURAL as the number of packets forwards to a neighbor in a certain period of time. Direct reputation is calculated based on the direct relationship between two agents. Direct reciprocity is related as the feedbacks between agents (I.e. nodes a_i and a_j) based on previous experiences. The current behavior of a neighbor (i.e node a_j) is monitored by the receptor node (i.e node a_i) calculating a current reputation ($R_{j,i}^{curr}$) as

$$R_{j,i}^{curr} = \frac{P_{FWD}}{P_{SENT}} \tag{7}$$

The current reputation denotes a fraction of the number of packets correctly delivered by the neighbor node a_j during a period of time. After this monitoring, the Trust Management (TM) computes the trust value by the sum between the direct reputation module (DRM) and the indirect reputation module (IRM). IRM refers to an expectation that the node a_i builds from its neighbor a_j involving another agent (i.e.a_p) which belong to the same neighbor domain. The IRM remains outside of the goal of this paper. Thus, the *reputation* value is computed taking only the DRM. The Trust Management calculates the direct reputation ($R_{j,i}^{D}$) by the sum of the current reputation $R_{j,i}^{curr}$ and a previous reputation $R_{j,i}^{prev}$. The value of $R_{j,i}^{prev}$ is obtained by the node a_i using the *history* table. The *direct* reputation is computed as

$$R_{j,i}^{D} = R_{j,i}^{prev} \cdot \Upsilon \ + \ R_{j,i}^{curr} \cdot (1 \ - \ \Upsilon) \tag{8}$$

Where the trust factor ($\Upsilon \in [0,1]$) permits the assignment of different importance degrees between the current reputation $R_{j,i}^{curr}$ and the previous ($R_{j,i}^{prev}$) reputation. The control action is executed when the direct reputation (i.e. $R_{j,i}^{D}$) falls below a critical threshold (λ). This means that the percentage of packet forwarded by the neighbor agent a_j is low, so that, the Route Discovery phase performs again to search a better path towards the destination.

IV. The Simulation Model

The network simulator *ns-2* [11] was used to simulate the NEURAL Agent described in section III. The simulation study was carried out by considering the "pause time" parameter. When the pause time (PT) increases, nodes tend to remain stationary. Otherwise, nodes are frequently in motion with low pause times (i.e full motion when $PT = 0$). To evaluate metrics, we consider the study of the packet delivery ratio and end-to-end delay such as in [12], [13], [14]. The packet delivery ratio is the fraction of the data packet received in the destination node. It is important to mention that for each configuration, reported measurements are the mean of at least 20 runs with different random seeds. The comparison study is carried out taking into account the Dynamic Source Routing (DSR) algorithm [10] and the Ad-Hoc On-Demand Distance Vector Routing (AODV) [15].

A. Comparison Study

In a base scenario, 50 nodes are randomly placed in a rectangular area of $300x1500\ m^2$. The maximum speed in the scenario is $20m.s^{-1}$ and the pause time is 30 seconds. The total length of the simulation is 900 seconds. Data traffic is generated by 20 constant bit rate (CBR) sources sending one $64 - byte$ packet per second. The transmission range is 300 meters, and the data rate is 2Mbits. In this first set of experiments the long side of the base area was extended between 1500 and 2500 meters. This produce that paths become longer and the network becomes sparser.

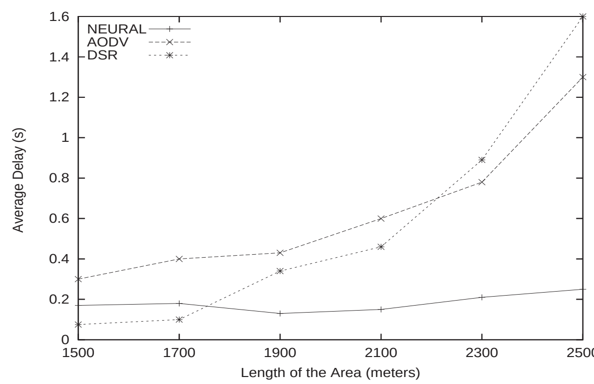

Fig. 2. Dense scenario: Delay

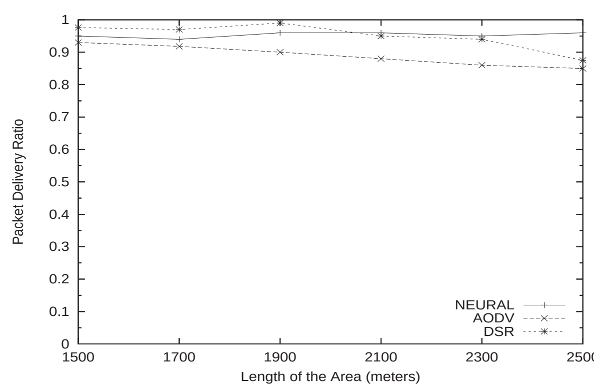

Fig. 3. Dense scenario: Delivery

The end-to-end delay is plotted in figure 2 and the packet delivery ratio in figure 3. Highly dynamic networks, where nodes are continuously moving, are the limiting situation for an Ad Hoc network and by far the most demanding scenario to be handled in this section. Under these conditions, the NEURAL protocol delivered over 95% packets for all the simulation scenarios, while DSR and AODV show a decreased behavior as the network becomes sparser. In particular, AODV always had the large end-to-end delay and the low packet delivery ratio. In the same way, the end-to-end delay was drastically increased for DSR in large areas ($> 300x1900\ m^2$). However, the packet delivery ratio oscillated between 0.98 and 0.99 for small areas ($\leq 300x1900\ m^2$). Considering networks with 50 nodes in an area $\leq 300x1700\ m^2$, the DSR protocol outperforms the others protocols inclusive with low end-to-end delay ($< 0.13\ s$). For example, the base configuration scenario ($300x1500\ m^2$) is the same such as in [13], and [14]. These results were quite similar based their simulations on small sized networks (50 nodes) and low traffic. In this paper, the number of nodes and the traffic load is heavily increased to evaluate the performance of DSR and AODV over disperse network. We observe that the increase of the area reduces the DSR performance in disperse networks with a large number of nodes and sparse connectivity. Moreover, DSR presented a huge end-to-end delay (until 1.6 seconds) for the largest scenario as well as the percentage of delivery packet falls until almost 10% for the largest configuration with $300x2500\ m^2$. In addition, NEURAL shows a constant behavior in term of the packet delivery ratio and low end-to-end delay as the area increases.

In a second set of experiments, the last dimension from the previous simulations was kept on $300x2500\ m^2$. In this case, the mobility of the nodes was varied considering the following pause times: 0, 15, 30, 60, 100, 300, 600 and 900 seconds.

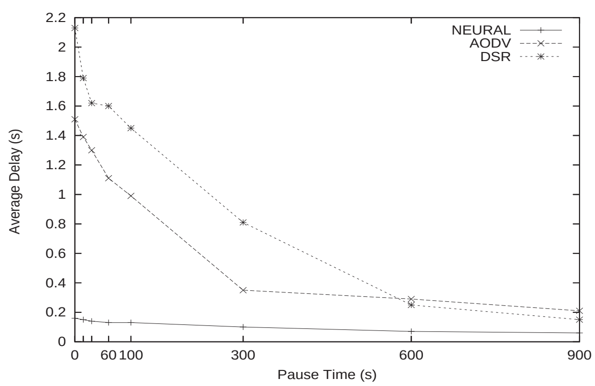

Fig. 4. Mobility: Delay

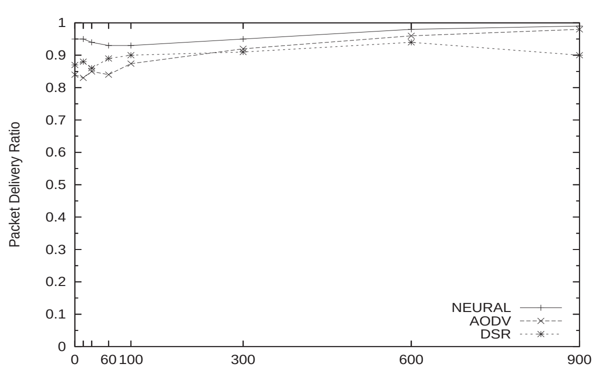

Fig. 5. Mobility: Delivery

With higher mobility configurations ($PT \leq 100\ s$), NEURAL is generally better than AODV and DSR with a packet delivery ratio greater than 93% for all the pause times such as shown figure 5. In the same way, the end-to-end delay plotted in figure 4 shows a latency under 0.2 seconds.

V. Conclusions

The main intention behind this paper was the development of an intelligent agent within the context of MANET, by applying bioscience, and cross-layer design concepts. The modeling of cross-layer frameworks was carried out to enhance the NEUron Routing ALgorithm (NEURAL) performance for Ad Hoc networks. In comparison with DSR and AODV, the packet delivery ratio and end-to-end delay were considerably improved taking into account different simulation topologies.

References

[1] V. E. Mujica-V., D. Sisalem, and R. Zeletin-Popescu, “Neural: A self-organizing routing algorithm for ad hoc networks,” in *Proceeding of 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Trentino, Italy, April 2005.

[2] D. C. G., F. Ducatelle, and L. Gambardella, “Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *European Transactions on Telecommunications*, vol. 16, no. 2, 2005.

[3] F. Ducatelle, D. C. G., and L. Gambardella, “Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks,” *International Journal on Computational Intelligence and Applications (IJCIA)*, vol. 16, no. 2, to appear in 2005.

[4] R. Choudhury, K. Paul, and S. Bandyopadhyay, “Marp: A multi-agent routing protocol for mobile wireless ad hoc networks,” *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 1, pp. 47–68, January 2004.

[5] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 69–74, 2004.

[6] C. Bishop, *Neural Network for Pattern Recognition*. Great Britain: Oxford University Press, 2000.

[7] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[8] H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps- An Introduction*. New York: Addisonn-Wesley, 1992.

[9] Y. Rehabi and V. E. Mujica-V., “A reputation-based trust mechanism for ad hoc networks,” in *Proceedings of the Tenth IEEE Symposium on Computers and Communications*, Cartagena, SPAIN, june 2005.

[10] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353. [Online]. Available: citeseer.nj.nec.com/johnson96dynamic.html

[11] U. Berkeley and U. ISI, “The network simulator ns-2,” 1998, available from http://www.isi.edu/nsnam/ns.

[12] Z. Hass and M. Pearlman, “The performance of query control schemes for the zone routing protocol,” in *IEEE/ACM Transactions on networking*, vol. 9 N4, Aug. 2001, pp. 427–438.

[13] C. Perkins, E. Royer, S. Das, and M. Marine, “Performance comparison of two on-demand routing protocols for ad hoc networks,” in *IEEE Personal Communications*, vol. 8 N1, february 2001, pp. 16–28.

[14] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc networks routing protocols,” in *Fourth annual ACM/IEEE international conference on Mobile computing and networking*. ACM Press, 1998, pp. 85–97.

[15] C. Perkins and M. Royer, “Ad-hoc on-demand distance vector routing,” *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, pp. 90–100, Feb. 1999.