

OPTIMIZATION OF EIGRP DYNAMIC ROUTING PROTOCOL BASED ON ARTIFICIAL INTELLIGENCE ALGORITHM

James Kouawa Tamgno

DETIC Dept.
ESMT Dakar, Sénégal.
james.kouawa@esmt.sn

Emile Kone Araba

Telecoms Engineer
ESMT Dakar, Sénégal
arabaemilek@gmail.com

Zacharia Damoue

DETIC Dept.
ESMT Dakar, Sénégal
zacharia.damoue@ucad.sn

Claude LISHOU

ICT Researcher
ESP/UCAD Dakar, Sénégal
claudio.lishou@ucad.sn

Abstract - The objective of this work is to propose an algorithmic approach to deal with the problem of EIGRP routing in a communication network. This type of network is characterized by dynamic traffic conditions, which requires taking into account the QoS at the routing level. Therefore, any adaptive routing approach must be sufficiently responsive and robust to account for any changes in traffic conditions while minimizing end-to-end routing time. However, the effectiveness of this approach is highly dependent on load information. This information must be sufficient and relevant and at the same time reliably reflect the actual network load at the time of the routing decision

Keywords – EIGRP; Artificial Intelligence; Routing; Optimization; QoS.

I. INTRODUCTION

As the routing problem has become more and more crucial in a computer network, we are witnessing a strong growth in data today. Nowadays, the increasingly complex multimedia services transmitted by telecommunication networks are present. We are now finding applications that entrust the network with more sensitive data such as voice, image or urgent information. The use of new generations of networks in multimedia applications or services with guaranteed quality, broadcasting and mobile services requires that routing be carried out with controlled QoS. For efficient routing management within a network, QoS must be guaranteed.

Despite the many routing algorithms within the network, in particular the DUAL algorithm for the EIGRP, which is one of the best algorithms to make a network flexible and optimal. Currently, the evolution of data flows is challenging this routing algorithm.

A. Context and motivations

1) Issues of quality of service of a 3G network

Thus, routers despite current routing algorithms find it difficult to manage and correctly route these huge data flows. From this perspective, the current routing is obliged to have new intelligent functionalities in order to be able to meet users' needs and provide a better quality of service. Thus, the use of new generations of networks in multimedia applications or services with guaranteed

quality requires that routing be ensured with a controlled Quality of Service (QoS).

2) Methodological approach

Objectives

- Study and analyze the limits of the DUAL algorithm and the EIGRP routing protocol;
- Study the various algorithms of artificial intelligence;
- Analyze and find a method for integrating artificial intelligence to optimize or improve routing performance;
- Develop and test the performance of the new routing algorithm;

B. Analyses of EIGRP performance and limitations

The EIGRP routing protocol is a protocol with many advantages and limitations. Some limits are directly visible through its cost formula used and others are seen during implementation in a simulation environment.

When the protocol is implemented, it establishes links with all routers on the network through its algorithm. One of the things we did was that once the protocol was configured it allowed for rapid convergence, but when an intermediate path (considered to be part of the best path) is cut off, communication between the algorithm and its protocol takes a long time to realize a change in topology. Due to this delay when we send a package, it is delayed before arrival at its destination or simply lost.

As a result, the algorithm was delayed in updating and informing the protocol of a change in topology.

The second point is that the default protocol in its general formula is composed as such [1]:

$$EIGRP \text{ Metric} = [k1 * BW + (k2 * BW) / (256 - LOAD) + k3 * DLY] \quad (1.1)$$

And if the coefficient k5 is non-zero, then the metric must be corrected as follows [1]:

$$EIGRP\ Metric = EIGRP\ metric * \left(\frac{k5}{RLY} + K4 \right) \quad (1.2)$$

By default the protocol does not consider the load. So when sending a packet the protocol only knows the size of the packet and the bandwidth used. Failure to take the load into account can cause packet congestion or delays in routing.

From these different points we can enumerate some disadvantages:

- Possibility of packet loss due to a change in topology;
- Possibility of rejecting or losing packets because the load is not taken into account when calculating the costs on a link;
- Increased routing time when deleting an intermediate link, (part of the path taken to arrive at a destination);
- Consumption of system resources when recalculating the algorithm.

So the protocol does not allow the respect of certain QoS parameters. Routing must then take into account the actual state of the links and routers in order to perform routing by quality of service.

Routing must then take into account the actual state of the network, information at the router level and the determination of routes to meet service quality constraints.

II. ANALYTICAL APPROACH

A. Optimization through an Artificial Intelligence approach

In the quest for an algorithmic approach based on Artificial Intelligence we have gone through some steps. We made a meticulous analysis of some Artificial Intelligence algorithms: Linear Regression, Linear Discriminant Analysis, Media Vector Machine, Nearest K-neighbors, Decision Tree and Neural Networks.

The study and comparison of its different algorithms was based on certain criteria such as: the resolution of an optimization problem which consisted in finding an algorithm capable of solving all types of problems (linear and non-linear), the consideration of several constraints at the input level (these input data can be discrete or continuous) and the calculation time of the algorithm which must be fast allowing to process several data in a fast and coherent way.

Following these different criteria, we have chosen as an algorithm the neural networks that have proven to be efficient and comply as closely as possible with all the required criteria.

The type of learning we have chosen is that of unsupervised learning, which is the one that most corresponds to our problem of developing the machine in an unknown environment that will make decisions based on a well-defined methodology within it to provide an optimal result.

We relied particularly on Hopfield neural networks. The Hopfield neural network is a model of a recurrent neural network whose connections matrix is symmetrical and null on the diagonal and where the dynamics is asynchronous (a single neuron is updated at each unit of time) [2]. It was popularized by physicist John Hopfield in 1982. His discovery rekindled the interest in neural networks that had run out of steam in the 1970s following an

article by Marvin Minsky and Seymour Papert [2]. Its features are as follows [3]:

- First networks for which proof of convergence has been established;
- First attempts to implement an RNA that takes into account the asynchronous nature of PSP and the fact that they have a duration in biological neurons;
- A continuous time variant of the network even tries to model membrane capacitance and axon resistance;
- First applications of an RNA to solve optimization problems.

One of the innovative contributions of the Hopfield network was the use of an auxiliary function, called the energy function or Lyapunov function, defined in terms of network variables, to demonstrate the convergence of the network towards a stable state.

Lyapunov's theory predicts that a dynamic system will have this behaviour if its auxiliary function decreases during the transient phase of the system and then stabilizes.

The energy function of this network was the innovation introduced to solve optimization problems.

To put his idea into practice, Hopfield modeled it as the following circuit:

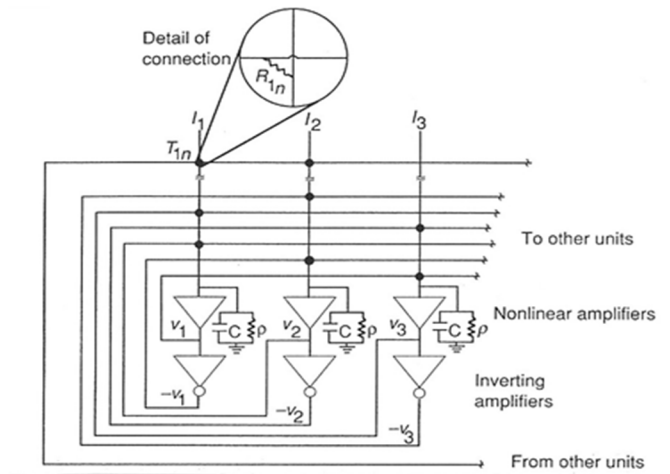


Figure 1 : Modeling as a circuit [3]

The network is modelled by an electrical circuit composed of a set of amplifiers that have very high input impedances and are grouped in pairs, the second element of which is a polarity inverter. The first element performs the output function and has a resistor and capacitor connected in parallel to the input. These simulate membrane effects in a biological neuron. In addition, the outputs of the other neurons that feed it are connected to it through resistors, thus simulating the attenuation of PSP in the dendritic tree. The circuit also includes a set of inputs that are simulated by currents I [B1].

The application of Kirchhoff's law on the currents at the input of an amplifier i gives [3].

$$I_{net} + \frac{u_i}{\rho} + C \frac{du_i}{dt} = \sum_{j=1}^m \frac{v_j - u_i}{R_{ji}} + I_i \quad (3.1)$$

As the term I_{net} is very low (due to the very high input impedance of amplifiers), we can ignore it. The equation then becomes, by rearranging the terms and posing [3],

$$\frac{1}{R_i} = \frac{1}{\rho} + \sum_{j=1}^m \frac{1}{R_{ji}} \quad (3.2) \text{ et } T_{ji} = \frac{1}{R_{ji}} \quad (3.3)$$

$$C \frac{du_i}{dt} = \sum_{j=1}^m T_{ji} v_j - \frac{u_i}{R_i} + I_i \quad (3.4)$$

For a network composed of m neurons, we obtain m equations of the form from above, whose solution describes the evolution in time of the total network. In each equation, u represents the activation level of a neuron, and v is its output, and T_{ij} acts as a synaptic weight. Moreover, R_i is the resistance equivalent to all the resistors present at the input of the unit i and C and the capacitance of the unit; the product $\tau = RiC$ is the time constant of the unit [3].

To prove the truth of his theory Hopfield applied it to the legendary problem of the commercial traveler [3]. The problem can be summarized as follows :

Find a closed path that

- Crosses n cities without ever revisiting one of them;
 - Is the shortest, if possible.
- A classic optimization problem whose complexity comes from the number of paths to consider when we approach it head-on.
- For n cities there are $n!$ possible paths.

The solution used by Hopfield is: Use a square network composed of $n \times n$ neurons and such that each column identifies a city in the path at a given time (state) [3].

For example, for a circuit comprising 5 cities, one possible solution is:

Table 1 : Commercial traveller [3]

E1	E2	E3	E4	E5	
0	1	0	0	0	V1
1	0	0	0	0	V2
0	0	0	0	1	V3
0	0	0	1	0	V4
0	0	1	0	0	V5

All these requirements can be met, in addition to the wish for a minimum length path, if we start from the following energy function, defined by Hopfield and Tank (the idea being that the minimum of the function will correspond to a network activation state that solves the problem).

$$E = \frac{A}{2} \sum_{X=1}^n \sum_{i=1}^n \sum_{j=1}^n v_{Xi} v_{Xj} + \frac{B}{2} \sum_{i=1}^n \sum_{X=1}^n \sum_{Y=1}^n v_{Xi} v_{Yi} + \frac{C}{2} \left(\sum_{X=1}^n \sum_{i=1}^n v_{Xi} - n \right)^2 + \frac{D}{2} \sum_{X=1}^n \sum_{Y=1}^n \sum_{i=1}^n d_{XY} v_{Xi} (v_{Y(i+1)} + v_{Y(i-1)}) \quad (3.5)$$

The indices X and Y refer to cities (the rows of the matrix), the indices i and j refer to the position of a city in the tower (the columns of the matrix), A , B , C and D are problem dependent constants (and are independent of the city names in the previous example) and d_{XY} is the distance between two cities [3]. The analysis of the different terms shows that the first is nil if each city is visited only once, that the second is nil if two cities are never visited simultaneously and that the third and also nil if the sum of the 1 is exactly n , the last one adds up the distances travelled from city to city and does not cancel each other out [3]. From these stated facts he had to establish the synaptic weights. For the n^2 neurons in the network, n^4 synaptic weights are required, chosen so as to respect the shape of the energy function. The latter were defined using an inhibitory approach, with each neuron trying to block the other neurons on its line and column [3].

$$T_{Xi,Yj} = -A\delta_{XY}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{XY}) - C - Dd_{XY}(\delta_{j(i+1)} + \delta_{j(i-1)}) \quad (3.6)$$

If we take the equation that governs neural activation, we have:

$$C \frac{du_i}{dt} = \sum_{j=1}^m T_{ji} v_j - \frac{u_i}{R_i} + I_i \Rightarrow \frac{du_i}{dt} = \sum_{j=1}^m \frac{T_{ji}}{C} v_j - \frac{u_i}{R_i C} + \frac{I_i}{C} \quad (3.7)$$

As each term of $TX_i Y_j$ dépend d'une constante, depends on a constant, the constant C can be integrated without loss of generality. The same reasoning applies to variable I . The equation can be rewritten:

$$\frac{du_i}{dt} = \sum_{j=1}^m T_{ji} v_j - \frac{u_i}{\tau} + I_i \quad (3.8)$$

Where τ is the time constant of the neuron (assuming that R_i is constant). The equation takes the following form if we replace synaptic weights with their expressions and identify each neuron by its coordinates in the network (row-column) [3]:

$$\frac{du_{Xi}}{dt} = -\frac{u_{Xi}}{\tau} - A \sum_{j=1}^n v_{Xj} + B \sum_{Y=1}^n v_{Yi} + C \left(N - \sum_{X=1}^n \sum_{i=1}^n v_{Xi} \right) + D \sum_{Y=1}^n d_{XY} v_{Xi} (v_{Y(i+1)} + v_{Y(i-1)}) \quad (3.9)$$

The approximation $\frac{du_{Xi}}{dt} \approx \frac{\Delta u_{Xi}}{\Delta t}$ then solves the equation by iteration. Starting from

$$\Delta u_{Xi} = \left(-\frac{u_{Xi}}{\tau} - A \sum_{j=1}^n v_{Xj} + B \sum_{Y=1}^n v_{Yi} + C \left(N - \sum_{X=1}^n \sum_{i=1}^n v_{Xi} \right) + D \sum_{Y=1}^n d_{XY} v_{Xi} (v_{Y(i+1)} + v_{Y(i-1)}) \right) \Delta t \quad (3.10)$$

We will have, at each step

$$u_{Xi}(\text{nouv.}) = u_{Xi}(\text{préc.}) + \Delta u_{Xi} \quad (3.11)$$

avec

$$v = g(\lambda u_{Xi}) = \frac{1}{2} (1 + \tanh(\lambda u_{Xi})) \quad (3.12)$$

We will now see the algorithm for solving the case of the commercial traveller [3].

- 1) *Initialize neuron activations (use the number of cities divided by the number of neurons, plus a small random term for each unit)*

Repeat:

- a) *Shooting a neuron at random;*
- b) *Calculate the activation of the corresponding neuron:*

$$u_{Xi}(\text{nouv.}) = u_{Xi}(\text{préc.}) + \left(-\frac{u_{Xi}}{\tau} - A \sum_{j=1}^n v_{Xj} + B \sum_{Y=1}^n v_{Yi} + C \left(N - \sum_{X=1}^n \sum_{i=1}^n v_{Xi} \right) + D \sum_{Y=1}^n d_{XY} v_{Xi} (v_{Y(i+1)} + v_{Y(i-1)}) \right) \Delta t$$

$$v_{Xi} = 0.5(1 + \tanh(\lambda u_{Xi}))$$

- c) *Use the new value of Xi the rest of the iteration;*

Until the activations no longer change from one iteration to another.

In an example of 10 cities, Hopfield and Tank used the following parameter values: $A=B=500$, $C=200$, $D=500$, $N=15$ et $\lambda=50$.

The results of the algorithm were satisfactory because they were able to find the most optimal path (the path that most respects all constraints).

Despite its effectiveness, the Hopfield algorithm with its countless variables and constraints often remains difficult to implement and requires real improvements.

Thus these models are quite complex and are not always optimal even if the speed of execution is always present, it often resides that the results may not be satisfactory and require real improvements to be accepted as a solution, it is thus that improvements by Ali and Kamoun will be made to Hopfield's work by introducing many more constraints and providing more detail. In the next part we will see how the improvements brought by Ali and Kamoun can help solve an optimization problem in a telecommunication network including that of EIGRP.

B. Design and Modeling of the algorithmic approach based on Artificial Intelligence

The first part is the modification of the cost. This change in costs will allow us to take into account several factors.

Based on the EIGRP routing protocol, we can consider that the cost formula could be adapted as follows [4] :

$$\text{Cost} = K1 \times \frac{256 \times 10^7}{\text{bandwidth}} + K2 \times \frac{(256 \times \frac{10^7}{\text{bandwidth}})}{256 - \text{load}} + K3 \times \left(256 \times \frac{\text{delay}}{10} \right) \quad (4.1)$$

The K values are as follows:

$K1 = 1, K2 = 1$ et $K3 = 1$.

bandwidth : the lowest bandwidth, **load** : load, **delay** : sum of the delays of each link from source to destination.

The purpose of introducing the charge is not to overload a link. Thus we will prefer to use a path with the lowest cost, the lowest load and whose difference between bandwidth and load is capable of supporting the packet.

The second part consists in the modeling of our node in the network.

Starting from the models of the Hopfield neural networks we modelled a network node as the following figure 2:

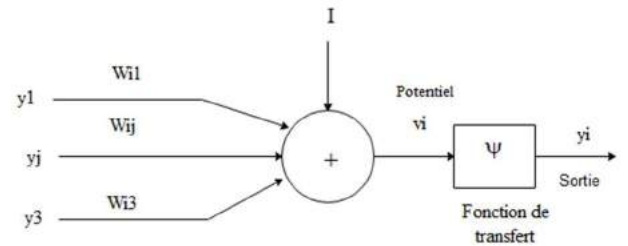


Figure 2 : Neural network model

$y1, y2, y3$: the inputs of the neuron, Wij : connection weight between neuron i and j , I : the direct input of the neuron, vi : Potential of the neuron and yi : the output.

The node receives information from the other nodes (y) with their characteristics (bandwidth, load, delay, etc.). To make a decision, the node uses our algorithm to make the optimal decision. During the decision making process, the output function will allow the activation of the link being considered as the most optimal in terms of characteristics (load, bandwidth, delay,...).

In addition to the inputs, we provide an input I that will allow the desired constraints to be respected for the proper functioning of the network. This variable I is the one inspired by Ali and Kammoun.

To put this input I we will go through some steps. First we will use the same activation function as Hopfield [5].

$$g_i(u_i) = \frac{1}{1 + e^{-a_i u_i}} \quad (5.1)$$

u_i The neuron input.

The coefficient a_i determines the inclination of the sigmoid function. The activation function $g(u_i)$ is similar to the threshold function for large values of a_i .

The stability of the Liapunov network can be verified by observing the energy function describing the state of the network [5]:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N T_{ij} v_i v_j - \sum_{i=1}^N I_i v_i \quad (5.2)$$

T_{ij} represents the synaptic weights, v_i the output of the neuron.

The change of energy, due to the change of state of the neuron i, is

$$\frac{\partial E}{\partial v_i} = -\sum_{j=1, j \neq i}^N T_{ij} v_j - \sum_{i=1}^N I_i \quad (5.3)$$

The state equation can be rewritten as follows:

$$C_i \frac{du_i}{dt} = -\frac{u_i}{R_i} - \frac{\partial E}{\partial v_i} \quad (5.4)$$

The Hopfield energy function modified by Ali Kamoun is described as follows [6]:

$$E = \frac{\mu_1}{2} \sum_{X=1}^n \sum_{\substack{i \neq X \\ (X,i) \neq (d,s)}}^n C_{Xi} v_{Xi} + \frac{\mu_2}{2} \sum_{X=1}^n \sum_{\substack{i \neq X \\ (X,i) \neq (d,s)}}^n \rho_{Xi} v_{Xi} + \frac{\mu_5}{2} (1 - v_{ds}) \\ + \frac{\mu_3}{2} \sum_{X=1}^n \left(\sum_{i \neq X}^n v_{Xi} - \sum_{i \neq X}^n v_{iX} \right)^2 \\ + \frac{\mu_4}{2} \sum_{i=1}^n \sum_{i \neq X}^n v_{Xi} (1 - v_{Xi}) \quad (6.1)$$

The constraints on variables and coefficients are as follows [6]:

$$\rho_{Xi} = \begin{cases} 1 & \text{si } l'arc(X, i) \text{ n'est pas dans le chemin} \\ 0 & \text{sinon} \end{cases} \quad (6.2)$$

$$v_{Xi} = \begin{cases} 1 & \text{si } l'arc(X, i) \text{ est dans le chemin} \\ 0 & \text{sinon} \end{cases} \quad (6.3)$$

The coefficients C_{Xi} are the connection costs of a node X to a node i.

μ_1 minimizes the total cost.

μ_2 prevents non-existent links from being included in the chosen path.

μ_3 is equal to zero for each node of the valid path (the number of incoming links is equal to the number of outgoing links).

μ_4 forces the state of the neural network to converge to one of the stable states.

μ_5 is zero when the v_{ds} at the output is equal to 1. This term is introduced to ensure that the source and destination nodes belong to the solution (the shortest path).

In the end, our entry I can be expressed as follows to be able to meet all our desired constraints [6].

$$I_{Xi} = -\frac{\mu_1}{2} C_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_2}{2} \rho_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2} \delta_{Xd} \delta_{is} \\ + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})] (1 - \delta_{Xd} \delta_{is}) \quad (5.1.8)$$

$$= \begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2}, & \text{pour } (X, i) = (d, s) \\ -\frac{\mu_1}{2} C_{Xi} - \frac{\mu_2}{2} \rho_{Xi} - \frac{\mu_4}{2} + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})] & \text{sinon} \end{cases} \quad (5.1.9)$$

The term δ_{ij} is Kronecker's term $\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$

For routing with packet loss reduction. Namely, each link is characterized by its ability [6], K_{ij} .

Each link has its own density traffic, G_{ij} , describing the actual traffic flow in a given direction. The traffic density certainly can not exceed the capacity of the link [6].

For this to be satisfied it is necessary that $K_{ij} \geq G_{ij}$. If, under certain circumstances, incoming packets overflow the input buffer, those packets would be lost. In order to minimize this effect, we introduce a new term observing free space in a binding capacity. This term is the difference $(K_{ij} - G_{ij})$ and the routing must find a path with enough freedom of space in link capacity. Since the neural network is designed to minimize the energy function, the real term in the energy function must be of the form $[1 - (K_{Xi} - G_{Xi})]$ [6]. As a result of this problem a new term, μ_6 has been introduced in the biases.

The term μ_6 minimizes the risk of packet loss under constraint $K_{ij} \geq G_{ij}$. Si G_{ij} exceeds the link capacity, K_{ij} , this router will be removed from the algorithm [6].

We obtain the following formula by replacing I with its value [6].

$$C_i \frac{du_i}{dt} = -\frac{u_i}{R_i} - \frac{\mu_1}{2} C_{Xi} (1 - \delta_{Xd} \delta_{is}) - \frac{\mu_2}{2} \rho_{Xi} (1 - \delta_{Xd} \delta_{is}) \\ - \mu_3 \sum_{Y \neq X} (v_{XY} - v_{YX}) + \mu_5 \sum_{Y \neq X} (v_{iY} - v_{Yi}) \\ - \frac{\mu_4}{2} (1 - 2v_{Xi}) + \frac{\mu_5}{2} \delta_{Xd} \delta_{is} \\ + \frac{\mu_6}{2} [1 - (K_{Xi} - G_{Xi})] (1 - \delta_{Xd} \delta_{is}) \quad (6.4)$$

From these different established formulas we have elaborated our algorithm by replacing the energy function of Hopfield by that which we have established. Then we re-selected the u_{Xi} that met the criteria to update the cost. Note that the algorithm must know the different paths to choose the optimal path output.

This is schematised as follows

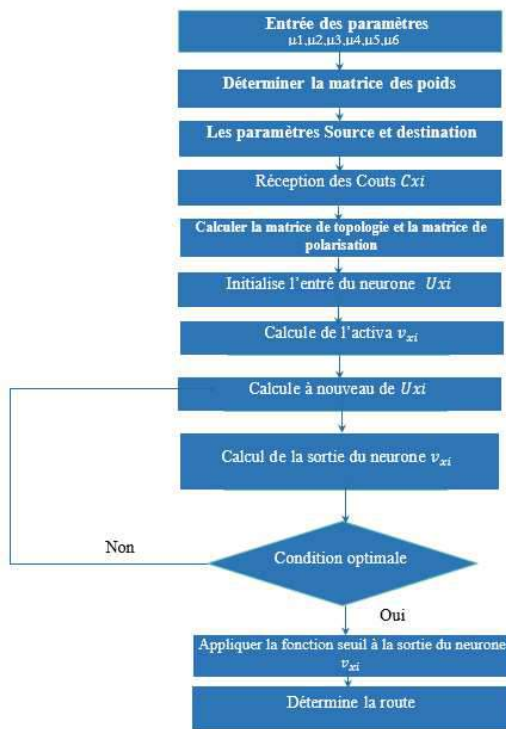


Figure 3 : Diagram of the Hopfield algorithm for routing in communication networks.

This diagram simulates how our algorithm works. At the beginning we have the input parameters, namely the multiplication coefficients allowing cost minimization, then we have the weight matrix, the source and destination simulation parameters, the cost consideration at the link level, the calculation of the topology and polarization matrix I , the initialization of the neuron input, and the calculation of the output. Then the algorithm recalculates the input neuron U_{xi} and finally calculates the output v_{xi} . If the optimal condition is reached, i.e. the activations no longer change, it applies the output and determines the associated route for us, otherwise it recalculates the U_{xi} input of the neuron again and continues until our satisfaction condition is met..

After these different elaborated formulas we built our 5 node network topology and then performed our tests. For the implementation of the algorithm it was realized in java.

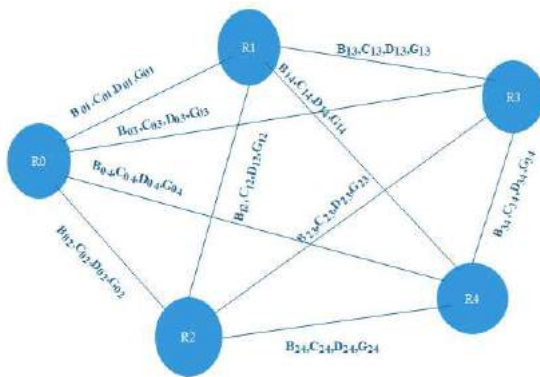


Figure 4 : Network topology

B indicates the bandwidth, C the cost, D the time and G the load.

Table 2 : Bandwidth

	R0	R1	R2	R3	R4
R0	1.024	1.544	1.024	1.024	1.544
R1	1.544	1.544	1.024	1.544	1.544
R2	1.544	1.544	1.544	1.544	1.544
R3	1.544	1.024	1.024	1.544	1.544
R4	1.544	0.0064	1.544	1.544	1.544

The values were chosen according to the bandwidths used by the EIGRP protocol with serial interfaces ranging from 0.0064 Mb to 1.544 Mb [7].

For the load we have the following table:

Table 3 : Loads

	R0	R1	R2	R3	R4
R0	0	80	75	50	85
R1	50	0	50	78	100
R2	80	50	0	120	78
R3	120	78	120	0	20
R4	120	100	78	70	0

Load values were chosen according to the EIGRP protocol ranging from 0 to 255 [1]. In addition, the load reflects the amount of traffic using the link. The load is measured dynamically with a value between 0 and 255.

For the delay we took $20\mu s$ on all interfaces in accordance with the EIGRP protocol.

Then we used the cost formula to obtain the following table:

Table 4 : Costs

	R0	R1	R2	R3	R4
R0	0	2179451.71926 5191	3025812.1546 961325	3024135.9223 300973	2179727.1763 173044
R1	2178079.7826 85246	0	3024135.9223 300973	2179345.8694 766257	2180659.4924 936895
R2	2179451.7192 65191	2178079.78268 5246	0	2182222.4931 423347	2179345.8694 766257
R3	2182222.4931 423347	3021780.90766 82315	2176528.0751 0517	0	2177056.6435 4088
R4	2178079.7826 85246	4.03076102564 1026E8	2179345.8694 766257	2178945.2337 17756	0

Following these different tables and our topology we realized different tests to see the performances of our algorithm.

DISCUSSION AND SOME CHALLENGES OF THE APPLICATION AND THE PERFORMANCE ANALYSIS

Our first test will be to send a packet to see the path followed by our algorithm and analyze if this result is coherent or not.

- Sending a package

We send a package of 1000 bytes of R0 => R3 and here is the result of the algorithm.

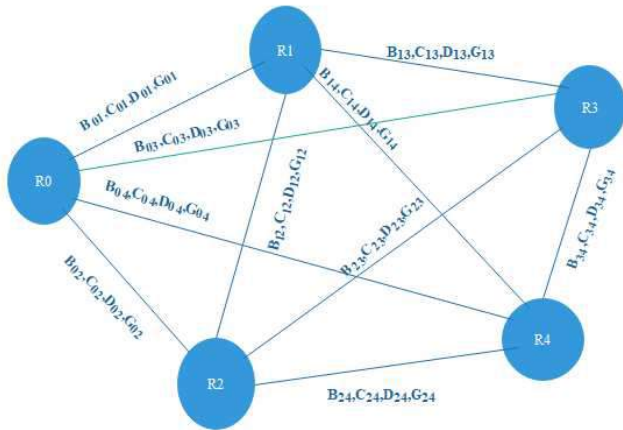


Figure 5.: Test1 result

The path followed by the algorithm is the one indicated in green it follows the direct path to reach the destination.

In the following cost table it is the black underlined path.

Table 5 : Result of costs

	R0	R1	R2	R3
R0	0	2179451.71 9265191	3025812.15 46961325	<u>3024135.92</u> <u>23300973</u>
R1	2178079.78 2685246	0	3024135.92 23300973	2179345.86 94766257
R2	2179451.71 9265191	2178079.78 2685246	0	2182222.49 31423347
R3	2182222.49 31423347	3021780.90 76682315	2176528.07 510517	0

In Table 5 he chooses the shortest way to go to the chosen destination. The algorithm could choose the following paths: R0=>R1=>R3 ou R0=>R2=>R3 and others, but among all these in terms of costs the path R0 =>R3 is the shortest. So the best in terms of costs. Routing time: 2 ms.

The bandwidth followed by the algorithm is described in the following table R0 =>R3.

Table 6: Results of the bandwidth

	R0	R1	R2	R3	R4
R0	1.024	1.544	1.024	<u>1.024</u>	1.544
R1	1.544	1.544	1.024	1.544	1.544
R2	1.544	1.544	1.544	1.544	1.544
R3	1.544	1.024	1.024	1.544	1.544
R4	1.544	0.0064	1.544	1.544	1.544

This bandwidth is the smallest possible in the table. The algorithm makes an optimal choice in terms of bandwidth as well. On the following table we can see the load of the link R0 =>R3.

Table 7: Results of the load

	R0	R1	R2	R3	R4
R0	0	80	75	<u>50</u>	85
R1	50	0	50	78	100
R2	80	50	0	120	78
R3	120	78	120	0	20
R4	120	100	78	70	0

The load of R0 => R3 is 50, which is the lowest in Table 7.

The algorithm is efficient in choosing the most optimal path in terms of costs, bandwidth and loads.

The algorithm had the possibility to choose other paths like R0 => R1 => R3 with a load of 80+78 = 158 which is totally higher than the load R0 => R3 as well as other paths were possible but are also not optimal compared to R0 => R3 in terms of costs, bandwidth and charges.

This optimal choice is specific to the algorithm because it takes into account all the factors that can occur when sending a packet. A choice not being known at all by DUAL because this by default only knows the bandwidth required to send the packet. While the bandwidth may be sufficient for a fast transmission of a packet provided that no other load is on this link at this time.

As a result, the algorithm has just covered an essential defect of DUAL by paying attention to the smallest details on a link and other factors for an optimal choice ensuring the delivery of the package in the best conditions.

The AI algorithm is therefore efficient in sending a single packet by making an optimal choice.

We will approach the second test which consists of sending several packets and observe the reaction of the algorithm.

- Sending several simultaneous packets

For this test we proposed to send three packets respectively 1000 bytes, 1400 bytes and 1200.

The following figure describes the cost of the different paths followed by the packets.

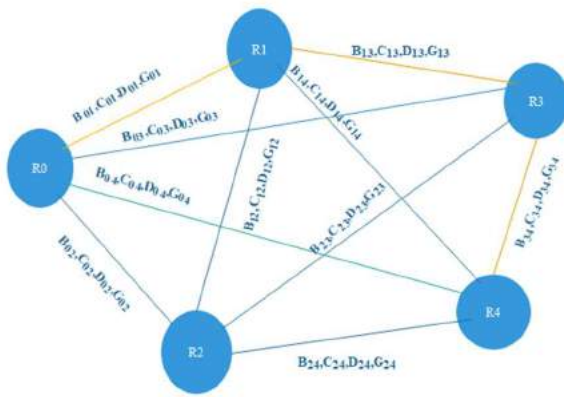


Figure 6 : Test2 result

The table summarizes the costs of the paths followed.

Table 8 : Test2 Result of costs

	R0	R1	R2	R3	R4
R 0	0	2179451.719265191	3025812.1546961325	3024135.9223300973	2179727.1763173044
R 1	2178079.782685246	0	3024135.9223300973	2179345.8694766257	2180659.4924936895
R 2	2179451.719265191	2178079.782685246	0	2182222.4931423347	2179345.8694766257
R 3	2182222.4931423347	3021780.9076682315	2176528.07510517	0	2177056.64354088
R 4	2182222.4931423347	4.030761025641026E8	2179345.8694766257	2178945.233717756	0

The path R0 => R1 => R3 => R4 in black, R0 => R4 in green.
The path R0 => R1 => R3 => R4 is certainly not the shortest in terms of costs, because other paths like R0 => R2 => R4 and R0 => R3 => R4 are better in terms of costs but this means that the AI did not make a choice based only on cost but on other parameters. Thus we will observe the following table to observe the choices of the AI in terms of bandwidth and loads.

Routing time: 5 ms.

The following table shows the loads at each node and the load monitored by the AI.

Table 9 : Test2 Results of the load

	R0	R1	R2	R3	R4
R0	0	80	75	50	85
R1	50	0	50	78	100
R2	80	50	0	120	78
R3	120	78	120	0	20
R4	120	100	78	70	0

Table 9 describes the loads followed by the algorithm.

The following table 10 describes the bandwidths monitored by the AI.

Table 10: Test2 Results of the bandwidth

	R0	R1	R2	R3	R4
R0	1.024	1.544	1.024	1.024	1.544
R1	1.544	1.544	1.024	1.544	1.544
R2	1.544	1.544	1.544	1.544	1.544
R3	1.544	1.024	1.024	1.544	1.544
R4	1.544	0.0064	1.544	1.544	1.544

The table describes the bandwidths followed by the algorithm.

For this test we sent three packages: the first package chose a different path from the last two packages. Depending on the number of packets in the queue to send, the algorithm chooses the number of paths available to send them. An optimal choice is made at each node to choose the best route or routes. By having different paths, the algorithm performs load balancing, to optimize routing time and reduce network overload.

On Table 9, the AI chooses for the first R0 => R1 => R3 => R4 the charge is then 80+78+20 = 178, there was another path R0 => R2 => R4 with the load 75+120 = 195, path R0 => R3 => R4 load: 50+20 = 70, path R0 => R4 85. We notice that the paths R0 => R4 and R0 => R3 => R4 were preferable. But when we look at the bandwidths used at the interfaces we see that the difference between the bandwidth and the load of the link R0 => R1 => R3 => R4 is greater than that of the link R0 => R3 => R4 which justifies the choice of the algorithm link. He therefore chooses the link R0 => R1 => R3 => R4 since it allows more fluidity and is optimal depending on the load, bandwidth and size of the packet. In the DUAL algorithm this would not be the case since the latter by default is based only on the cost and bandwidth of the link. It should be recalled that in the characteristics cited at the EIGRP level, it only shares the load according to cost and bandwidth, so once does not consider the load.

The AI is thus efficient in sharing loads not only on the basis of cost but also on other factors such as bandwidth and load.

- Sending a package with deletion of links

This test is a continuation of the first test. When the first packet was sent, it passed directly to the R3 node. Now we have removed this link to see if the package will take another path, would be lost or will take longer to reach its destination.

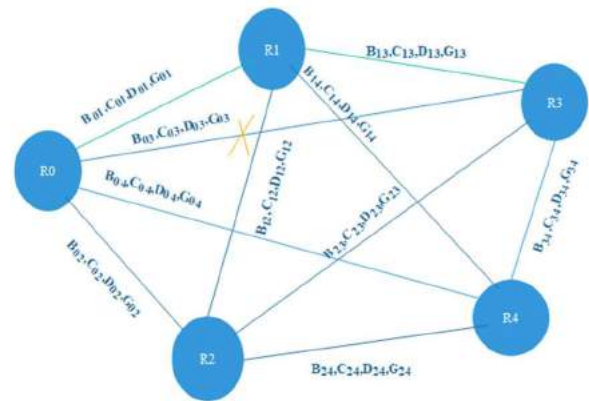


Figure 7: Topology of Test3

The path followed by the algorithm is indicated in green.

The following table shows the cost followed by the algorithm.

Table 5: Test3 Result of costs

	R0	R1	R2	R3	R4
R0	0	<u>2179451.71926</u> <u>5191</u>	3025812.1546 961325	~~3024135.9223~~ 300973	2179727.1763 173044
R1	2178079.7826 85246	0	3024135.9223 300973	<u>2179345.8694</u> <u>766257</u>	2180659.4924 936895
R2	2179451.7192 65191	2178079.78268 5246	0	2182222.4931 423347	2179345.8694 766257
R3	2182222.4931 423347	3021780.90766 82315	2176528.0751 0517	0	2177056.6435 4088
R4	2182222.4931 423347	4.03076102564 1026E8	2179345.8694 766257	2178945.2337 17756	0

The path followed by the algorithm at the cost level R0 => R1=> R3 after deleting the link R0 => R3.

Looking at the table we see that this path is the most optimal in terms of costs.

Routing time: 2 ms.

The following table describes the bandwidth followed by the algorithm with removal of the link R0 => R3.

Table 12 : Test3 Results of the bandwidth

	R0	R1	R2	R3	R4
R0	1.024	<u>1.544</u>	1.024	~~1.024~~	1.544
R1	1.544	1.544	1.024	<u>1.544</u>	1.544
R2	1.544	1.544	1.544	1.544	1.544
R3	1.544	1.024	1.024	1.544	1.544
R4	1.544	0.0064	1.544	1.544	1.544

The path followed by the algorithm at the bandwidth level.

In terms of bandwidth it is not the smallest because the path R0 => R2 => R3 has a smaller bandwidth.

The following table describes the load followed by the algorithm.

Table 13: Test3 Results of the load

	R0	R1	R2	R3	R4
R0	0	<u>80</u>	75	~~50~~	85
R1	50	0	50	<u>78</u>	100
R2	80	50	0	120	78
R3	120	78	120	0	20
R4	120	100	78	70	0

The table shows the path followed by the algorithm in terms of loads when the path R0 => R3 has been deleted.

When we look at the table we see that the AI takes into account the load because despite the bandwidth offered by the link R0 => R2 => R3 this is not optimal in terms of the load circulating on this link. This approach allows to see the introduction of the load because this can be decisive when choosing the optimal path. At the level of DUAL since it is based on cost and bandwidth, the latter would therefore make a sub-optimal choice, thus committing a risk of packet loss.

The result observed despite the removal of the link R0 => R3 is as follows:

The algorithm can find another way to route the packet, using the optimal bandwidth, the lowest cost and the smallest load with a reduced routing time. Despite the deletion of a path this does not have a very serious impact on the routing time. This shows the fast reaction of the path deletion algorithm being considered the best to find another one as quickly as possible to complete the packet. We can therefore say that the algorithm has a fast update time. Conclusion There is no packet loss and the algorithm manages to find another path to be able to route the packet without any real impact on the completion time.

In view of the results of the various tests performed, the advantages of the algorithm can be summarized as follows:

- Optimal path in terms of bandwidth, load, and cost;
- Load sharing taking into account the actual load on a link;
- Reduced packet loss;
- Reduced convergence time, even if in these different tests the convergence time may depend on the environment. Despite this fact, neural networks are especially known for their speed of information processing, those that confirm this last point.

III. CONCLUSION

At the end of this study, we propose to summarize our work, to analyze the results obtained and finally draw up perspectives that will make it possible to envisage new lines of research as a logical continuation of this work.

In this work, we presented a new algorithmic approach that allows better consideration of different states in routing decisions.

To support our approach, we performed a critical analysis on the EIGRP dynamic routing protocol and its performance with the parameters. This analysis led us to use as a basis for our work the algorithm of standard unsupervised learning neural networks with more advantages and better compliance with optimal conditions.

The AI could then be an improvement that could be made to the EIGRP routing protocol during the search phase for the most optimal path, sending a packet to take into account the load of a link and make efficient load sharing.

It turned out that the algorithm proved its efficiency in our various tests carried out taking into account certain parameters.

Further research may be conducted to improve the algorithm in another environment by taking into account other parameters that may be relevant when deciding on an optimal route in the network.

REFERENCES :

- [1]http://novamine.free.fr/root/COURS%20TCRT/Cours%20Re-seaux%20IP/Routage%20Statique%20%26%20Dynamique/Chapter6_EIGRP.pdf (Accessed June 18, 2019).
- [2]http://novamine.free.fr/root/COURS%20TCRT/Cours%20Re-seaux%20IP/Routage%20Statique%20%26%20Dynamique/Chapter6_EIGRP.pdf (Accessed June 19, 2019).

[3]https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_de_Hopfield (Accessed July 15, 2019).

[4] <https://www.info2.uqam.ca/DIC9310/Notes/6-DIC9310-BAM-Hopfield-BSB-> (Accessed July 24, 2019).

[5] <https://www.info2.uqam.ca/DIC9310/Notes/6-DIC9310-BAM-Hopfield-BSB-> (Accessed July 24, 2019).

[6] Nenad Kojic, Irini Reljin et Branimir Reljin (Janvier 2006) “Dynamic Routing Algorithm based on Artificial Intelligence”, pages 188–193.

[7] Nenad Kojic, Irini Reljin et Branimir Reljin (Janvier 2006) “Dynamic Routing Algorithm based on Artificial Intelligence”, pages 188–193.

[8] J. J. Hopfield and D. W. Tank, (1985), “Neural computations of decision in optimization problems,”, pages 141–152.

[9] J. J. Hopfield and D. W. Tank, (1985), “Neural computations of decision in optimization problems,”, pages 141–152.

[10] Sushma Jain et J.D. Sharma (3 Juin 2010), “Delay Bound Multicast Routing Using Hopfield Neural Network”, pages 384–389.

[11] Sushma Jain et J.D. Sharma (3 Juin 2010), “Delay Bound Multicast Routing Using Hopfield Neural Network”, pages 384–389.

[12]<http://www.nyuplanet.eu/ccna/semestre3/course/module8/8.1.4.1/8.1.4.1.html> (Accessed July 24, 2019).



Dr. James Tamgno Kouawa, As a senior lecturer at ESMT, i obtained my Ph.D in may 2014 at University Cheikh Anta Diop of Dakar. My researches focused on Promotion of African's languages and dialects and Contribution to a Wider Accessibility of ICT, by Circumventing Language Barriers and Lessening the Impact of Illiteracy or Disability. Previously I have gotten a M.Sc. in Engineering Science from University Cheikh Anta DIOP of Dakar, studied numerical analysis and mathematics at the University of Yaoundé, and worked as a research scientist at ESMT (Multinational High School of Telecommunication) of Dakar. <http://www.esmt.sn>. IEEE-ID: 92230617

Emile Kone ARABA



Dr Zacharia DAMOUE is a teacher-researcher at the Multinational Higher School of Telecommunications (ESMT) in Dakar. He has done extensive work on the telecommunications access network part. As such, he has extensive experience in fiber optic and optical free space networks.



Pr Claude Lishou is currently a Professor with the Université Cheikh Anta Diop (UCAD), Dakar, Senegal. He is also teaching several disciplines at Ecole Supérieure Polytechnique ranging from industrial IT to next-generation networks (NGN). As the Director of the Virtual Platform at UCAD, he is responsible for the coordination of the IT system and steers the e-learning strategy for the entire university. He directs the Research and Development Laboratory, which is internationally renowned (Laboratory for Information Processing) and dedicated to the use of ICTE in education and training, environment, energy, transport, and e-governance. On a scientific level, he has authored or coauthored around 54 articles in scientific reviews, which have an international circulation. For more than two decades, he has supervised and given support to dozens of engineers and researchers preparing their theses in the sub-region. As a recognized expert in the development of ICTE applications and services, he divides his time between teaching, research, and consultancy with international organizations, such as OIF, AUF, UNDP, UNESCO, and UNCTAD. He is a member of several academic societies, editorial committees of scientific reviews, and Information and Communication Technology Networks for Teaching and Research. He is also the Editor-in-Chief of the Journal des Sciences Pour l'Ingénieur (Journal of Sciences for the Engineer). (Based on document published on 27 April 2020).