PERGAMON

Engineering Applications of Artificial Intelligence 14 (2001) 51–60

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

An artificial neural network approach for routing in distributed computer networks

Samuel Pierre[a,*], Hassane Said[b], Wilfried G. Probst[c]

[a] *Mobile Computing and Networking Research Laboratory (LARIM), Department of Electrical and Computer Engineering, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Qué, Canada H3C 3A7*

[b] *Canada Customs and Revenue Agency, 875 Heron Road # 3160, Ottawa, Ont, Canada K1A 0L5*

[c] *Department of Computer Science, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montreal, Que, Canada H3C 3P8*

Received 1 July 1999; received in revised form 1 June 2000; accepted 1 July 2000

Abstract

This paper presents a new approach based on the Hopfield model of artificial neural networks to solve the routing problem in a context of computer network design. The computer networks considered are packet switching networks, modeled as non-oriented graphs where nodes represent servers, hosts or switches, while bi-directional and symmetric arcs represent full duplex communication links. The proposed method is based on a network representation enabling to match each network configuration with a Hopfield neural network in order to find the best path between any node pair by minimizing an energy function. The results show that the time delay derived from flow assignment carried out by this approach is, in most cases, better than those determined using conventional routing heuristics. Therefore, this neural-network approach is suitable to be integrated into an overall topological design process of moderate-speed and high-speed networks subject to quality of service constraints as well as to changes in configuration and link costs. © 2001 Published by Elsevier Science Ltd.

Keywords: Computer networks; Flow assignment; Heuristic method; Neural network; Routing algorithm; Topological design

1. Introduction

A computer network is a communication infrastructure which consists of nodes representing workstations, servers, hosts, or switches, as well as arcs representing communication links. In a packet switching network, each message is broken into small blocks called packets, which travel independently through the network from the source node to the destination node. The nodes through which the packet is transmitted from the source to the destination constitute a path or a route, and the mechanism used to select one route among various alternatives to link each source–destination pair is called a routing procedure.

There exist in the literature many routing procedures (Kershenbaum, 1993). They essentially aim at minimizing the main delay of the packets in a network, that is, the average time taken by a typical packet to travel from one source to a given destination in the network. Among them, exact mathematical programming methods (Dutta and Mitra, 1993; Gavish, 1992; Neumann, 1992), based on certain properties of the mean delay function to solve the problem of optimal routing, are revealed themselves unpractical, particularly because of their incapacity to take into account possible node or link failures. Furthermore, their implementation requires complex and lengthy calculations. As a result, for practical purposes, routing heuristics are recommended in order to efficiently select the transmission paths of the packets without causing network congestion (Baransel et al., 1995; Beaubrun and Pierre, 1997; Kamimura and Nishino, 1991; Pierre and Beaubrun, 2000; Khasnabish, 1993), and neural networks have been considered as a general framework to solve such kind of optimization problem (Binh and Chong, 1995; Lee and Chang, 1993; Mehmet and Kamoun, 1993; Rauch and Winarske, 1988).

The brain's capacity for learning, as well as its resistance to local disruptions, essentially result from

*Corresponding author. Tel.: +514-340-4711 ext. 4685; fax: +514-340-3240.

E-mail address: samuel.pierre@polymtl.ca (S. Pierre).

0952-1976/01/$ - see front matter © 2001 Published by Elsevier Science Ltd.
PII: S0952-1976(00)00059-2

the collective functioning and simultaneous operation of the neurons of which it is composed. The human brain is made up of an enormous number of nerve cells called neurons, which are organized into a strongly connected network. Each neuron receives signals coming from several thousands of other like cells via branches called *dendrites*. The signal passes from one neuron to another by crossing a single fiber, called the *axon*, branching from its extremity. The point of contact between two neurons is called a *synapse*, which may be stimulating or inhibiting in nature. Each synapse is characterized by the efficiency with which it ensures the connection. The neuron bases its decisions on the global information it receives. A formal or artificial neuron is a model of a biological neuron possessing those properties (Drosen, 1994; Kagiantis and Papantoni-Kazakos, 1997; Rumelhart et al., 1984).

This paper proposes a heuristic approach based on Hopfield artificial neural networks to solve the routing problem in a design context. It is organized as follows. Section 2 presents the Hopfield neural network. Section 3 exposes the proposed adaptation of the Hopfield model to solve the routing problem. Section 4 specifies the implementation details and presents some simulation results.

2. The Hopfield neural network

A formal neuron is a binary element whose state is either $+1$ (active) or -1 (inactive). The neuron calculates the sum of its inputs, which are the outputs of the formal neurons to which it is connected. It contains n inputs adjusted by a weight and transfer function, which may be continuous or discontinuous. The value of each input is modulated by the corresponding synaptic efficiency weight, and the neuron takes a decision by comparing this sum with its own intrinsic threshold. Its state is equal to -1 if the sum is less than the threshold, and $+1$ if otherwise. All neurons take their decisions simultaneously, while taking into account the evolution of the global state of the network.

Two rules of learning are often used: the Widrow–Hoff's rule and the perceptron rule (Drosen, 1994; Rumelhart et al., 1984). According to the Widrow–Hoff's rule, corrections are made to the synaptic coefficients of all the neurons proportionally to the difference between the response obtained and the response desired (gradient algorithm). In the case of the perceptron rule, the correction is made by acting on the synaptic coefficients of the neurons giving an incorrect response. These two rules allow the network to carry out *a supervised learning* (Kagiantis and Papantoni-Kazakos, 1997).

Formal neural networks are only capable of solving simple problems of classification. For more complex problems, a solution consists of organizing the decision-making process into several stages, which correspond to utilizing a *network of several layers*. A generalization of the Widrow–Hoff's rule has given rise to the method of *Retro-propagation* of the gradient. It consists of imposing a configuration on the input neurons, observing the response of the network provided by the output neurons, and then recalculating the synaptic efficiencies so as to minimize the difference between the real and the desired responses by using a *gradient* method. The calculation is made layer-by-layer, from the output toward the input, hence the name retro-propagation.

A Hopfield network consists of n completely connected neurons (Hopfield, 1982, 1984; Hopfield and Tank, 1986). Each neuron has two possible states: $V_i = -1$ and $V_i = 1$. As shown in Fig. 1, the connection of neuron i to neuron j is denoted by T_{ij} and the total entry of a neuron i is equal to $\sum_j T_{ij} V_j$. The state of the system is characterized by the n V_i, and may then be represented by a word of n bits. The network has a dynamic functioning usually sequenced by a clock: $V_i(t)$ or V_i denotes the state of the neuron at instant t, and $V_i(t+1)$ the state of the neuron at instant t+dt, dt denoting the interval between two ticks of the clock.

The Hopfield model is relatively different from the aforementioned layer models. Firstly, the learning is static since there is no true dynamism in the connections. Secondly, the relaxation of the network is dynamic since the network is capable of making a certain number of iterations before reverting to a stable state.

Another aspect of Hopfield networks is their tendency to minimize an *energy function* which depends on the weight of the connections between the different neurons. This aspect raises the possibility of using the Hopfield neural networks in different types of applications, which require certain decisions to be taken in order to minimize a number of values in relation to other variables.

If the states of the network are $V_i \in [-1, +1]$, then the energy of a Hopfield and Tank (1986) network is

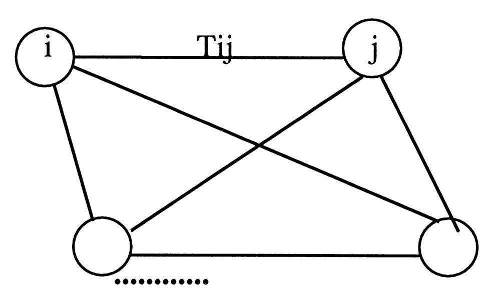


Fig. 1. Example of a Hopfield neural network.

defined as:

$E = -\frac{1}{2}S,$

where

$$S = V_1(w_{12}V_2 + w_{13}V_3 + w_{14}V_4 + \ldots) + V_2(w_{21}V_1 + w_{23}V_3 + w_{24}V_4 + \ldots) + v_3(w_{31}V_1 + w_{32}V_2 + w_{34}V_4 + \ldots) + \ldots$$

or

$S = V_1$(weight of the connections originating from neuron 1) $+V_2$(weight of the connections provided originating from neuron 2)$+\ldots$

More explicitly, E may be rewritten as follows :

$$E = -\frac{1}{2}\sum_{i,j} w_{i,j}v_i v_j.$$

Hopfield and Tank (1986) have proposed a circuit which can be viewed as a model of biological neural network; each neuron is represented as an operational non-linear amplifier with a sigmoid transfer function g_i. This monotone and increasing function relays the output V_i of neuron i to the entry U_i. The output V_i is designed to take any value between 0 and 1. A typical sigmoid function is (Mehmet and Kamoun, 1993)

$$V_i = g_i(U_i) = \frac{1}{1 + e^{-\lambda_i U_i}}, \tag{1}$$

where λ_i is the increase of the amplifier. Each neuron receives an input from the outside, and its connections from other neurons; the weights of these connections may be described by a connection matrix defining the neural network. The dynamics of Hopfield networks may be described as follows (Mehmet and Kamoun, 1993)

$$\frac{dU_i}{dt} = \sum_{j=1}^{n} T_{ij}V_j - \frac{U_i}{\tau} + I_i, \tag{2}$$

where τ denotes a circuit's time constant. For a symmetric connection matrix, and for a sufficiently high gain of the amplifiers ($\lambda_i \to \infty$), the dynamics of the neurons follow a decreasing gradient descent of the quadratic energy function E (Hopfield, 1984):

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ij}V_iV_j - \sum_{i=1}^{n} I_iV_j. \tag{3}$$

As long as the state of the neural network evolves inside a hypercube of dimension n defined by $V_i \in \{0,1\}$, the minimum of the energy function E will not attain one of the 2^n vertices of this hypercube, unless λ_i tends towards ∞. In terms of energy function, the dynamics of the ith neuron can be described by (Mehmet and Kamoun (1993)

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i}. \tag{4}$$

This relation constitutes the foundation of the proposed routing approach.

3. Routing by Hopfield neural networks

Routing in a packet switching network consists of determining the best path or route between each node pair (source/destination) through the network in order to minimize the network delay. Computer networks considered in this paper are modeled as non-oriented graphs defined by $G = (N, A)$, where N denotes the set of nodes and A the set of bi-directional and full-duplex links. The neural-network model proposed to solve the routing problem consists of $n(n-1)$ neurons, that is, a matrix $n \times n$ where all the neurons on the diagonal are eliminated. The coordinates of the neurons are (x, i), where x denotes the rows, and i the columns. The neuron at (x, i) is characterized by its output V_{xi} and defined as follows:

$$V_{xi} = \begin{cases} 1 & \text{if the arc } (x,\ i) \text{ is part of the route,} \\ 0 & \text{if not} \end{cases}$$

and the variable ρ_{xi} is defined as follows:

$$\rho_{xi} = \begin{cases} 0 & \text{if the arc } (x,\ i) \text{ exists,} \\ 1 & \text{if not.} \end{cases}$$

The cost of the arc (x, i) is denoted by C_{xi} which is a real positive variable. A null cost is assigned to each nonexistent arc. For the purpose of numeric manipulation associated with calculations of the derivatives and without loss of generality, the energy function proposed by Mehmet and Kamoun (1993) has been adopted

$$E = \frac{\mu_1}{2}\sum_{x=1}^{n}\sum_{\substack{i=1\\ i\neq x\\ (x,i)\neq(d,s)}}^{n} C_{xi}V_{xi} + \frac{\mu_2}{2}\sum_{x=1}^{n}\sum_{\substack{i=1\\ i\neq x\\ (x,i)\neq(d,s)}}^{n} \rho_{xi}V_{xi} + \frac{\mu_3}{2}\sum_{x=1}^{n}\left\{\sum_{\substack{i=1\\ i\neq x}}^{n} V_{xi} - \sum_{\substack{i=1\\ i\neq x}}^{n} V_{ix}V_{ix}\right\}^2 + \frac{\mu_4}{2}\sum_{i=1}^{n}\sum_{\substack{x=1\\ x\neq i}} V_{xi}(1 - V_{xi}) + \frac{\mu_5}{2}(1 - V_{ds}). \tag{5}$$

As the neurons are organized into a two-dimensional table, (1), (2) and (4) can be rewritten, respectively, as follows:

$$V_{xi} = g_{xi}(U_{xi}) = \frac{1}{1 + e^{-\lambda_{xi} U_{xi}}} \quad (6)$$

$$\frac{dU_{xi}}{dt} = \sum_{y=1}^{n} \sum_{\substack{j=1 \\ j \neq y}}^{n} T_{xi,yj} V_{yj} - \frac{U_{xi}}{\tau} + I_{xi} \quad (7)$$

$$\frac{dU_{xi}}{dt} = -\frac{U_{xi}}{\tau} - \frac{\partial E}{\partial V_{xi}}. \quad (8)$$

By substituting (5) in (8) and calculating the derivative $\partial E/\partial V_{xi}$, it follows:

$$\begin{aligned} \frac{dU_{xi}}{dt} = &-\frac{U_{xi}}{\tau} - \frac{\mu_1}{2} C_{xi}(1 - \delta_{xd}\delta_{is}) - \frac{\mu_2}{2}\rho_{xi}(1 - \delta_{xd}\delta_{is}) \\ &- \mu_3 \sum_{\substack{y=1 \\ y \neq x}}^{n} (V_{xy} - V_{yx}) \\ &- \mu_3 \sum_{\substack{y=1 \\ y \neq i}}^{n} (V_{iy} - V_{yi}) \\ &- \frac{\mu_4}{2}(1 - 2V_{xi}) + \frac{\mu_5}{2}\delta_{xd}\delta_{is}, \end{aligned} \quad (9)$$

where δ is the Kronecker symbol, which is defined as

$$\delta_{ab} = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{if not.} \end{cases}$$

In comparing the coefficients of (7) with those of (9), the $T_{xi,yj}$ of the connection matrix take the following values:

$$T_{xi,yj} = \mu_4 \delta_{xy}\delta_{ij} - \mu_3\delta_{xy} - \mu_3\delta_{ij} + \mu_3\delta_{jx} + \mu_3\delta_{iy} \quad (10)$$

$$\begin{aligned} I_{xi} = &-\frac{\mu_1}{2}C_{xi}(1 - \delta_{xd}\delta_{is}) - \frac{\mu_2}{2}\rho_{xi}(1 - \delta_{xd}\delta_{is}) \\ &- \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd}\delta_{is} \\ = &\begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2} & \text{if } (x,i) = (d,s), \\ -\frac{\mu_1}{2}C_{xi} - \frac{\mu_2}{2}\rho_{xi} - \frac{\mu_4}{2} & \text{if not,} \end{cases} \quad \forall (x \neq i), (y \neq j). \end{aligned} \quad (11)$$

The initial data of the neurons U_{xi} are equal to zero, and the evolution of the state of the neural network is simulated by the solution of a system of $n(n-1)$ differential equations where the variables are the neuron outputs V_{xi}. To solve this system, the numerical method of Runge–Kutta of the fourth order has been used. The solution consists of observing the outputs of the neurons V_{xi} for a specific duration δt. The circuit's time constant τ for each neuron is initialized to 1 and, without loss of generality, it has been considered that $\lambda_{xi} = \lambda$ and $g_{xi} = g$. Good results are obtained for δt ranged from 10^{-5} to 10^{-3}. To avoid bias in favour of any particular path, it must be assumed that all inputs U_{xi} are equal to 0. However, to help the network converge rapidly, while preventing it from adopting an undesirable state (for example, convergence of two different paths), small perturbations must be made to the initial inputs of the U_{xi} network. At the start and based on our simulation results, we have chosen U_{xi} such that $-0.0002 \leq U_{xi} \leq +0.0002$. The calculations will cease when the network reaches a stable state, that is, when the difference between the outputs is less than $10^{-5}(V_{xi} \leq 0.00001)$ from one update to another. When the network is in a stable state, the final values of V_{xi} are rounded off, that is, they are set to 0 if $V_{xi} < 0.5$, and to 1 otherwise.

The parameters μ_i of the neural network serve as regulators and precision factors to avoid blocking in a local minimum. Hopfield and Tank (1986) have used a formulation which combines mixed legality constraints. This solution, although simple, is far from efficient as much time is lost avoiding illegal states.

Mehmet and Kamoun (1993) have considered the problem in the form of an inequation system as follows:

$$\begin{aligned} &2\mu_3 - \mu_4 > 0, \\ &\mu_5 \gg \mu_1 (C_{xi})_{\max}, \\ &\mu_2 = \mu_5, \\ &\mu_1 \gg 2\mu_3/(C_{xi})_{\max}. \end{aligned}$$

Based on the fact the energy function is quadratic and the second derivative $\partial^2 E/\partial V_{xi}^2 > 0$, the following values have been found : $\mu_1 = 950$; $\mu_2 = 2500$; $\mu_3 = 1500$; $\mu_4 = 475$; $\mu_5 = 2500$.

This paper assumes that the μ_i are of the form μ_i= const.Vit, where const is an integer constant and Vit an integer variable, which takes values from 1 to 5. This variable serves as a speed and precision factor. Beyond the value of 5, the network begins to diverge and to give poor results. The mean number of updates needed to find results equals 100, compared to 6000 in the work of Mehmet and Kamoun (1993).

The first step of the proposed routing algorithm consists of obtaining the network data, that is, the number n of nodes, the matrix ρ of the links and the traffic matrix. The second step initializes the matrix of the V_{xi}: the V_{xi} receive random values between -0.0002 and $+0.0002$. The third step triggers the process of minimization in the neural network to solve the differential equations and to stabilize the network. Finally, the values of the matrix of outputs are rounded leading to the searched routing matrix. From this routing matrix, flow is assigned to each link of the network configuration by distributing the traffic between each node pair on the best route connecting these nodes : the flow of a link is the effective number of information unit (bit) carried by this link per unit of time (second). Then, a capacity value can be assigned to each link by using capacity options available on the marketplace, while ensuring that the link flow does not exceed the link capacity: the capacity of a link refers to the maximum number of information unit which can be transmitted on this link per second; as the link flow, it is usually expressed in bits per second (bps) (Jan et al., 1993).

Time delay is a function of the link flows and capacities. For moderate-speed networks subject to some realistic assumptions discussed in Gerla and Kleinrock (1977), the mean time delay T can be calculated as follows:

$$T = \frac{1}{\gamma}\sum_{i=1}^{m}\frac{f_i}{C_i - f_i}, \tag{12}$$

where C_i denotes the capacity of the link i, f_i the flow of the link i in bits/second (bps), γ the total traffic in the network (in packets/s), and m the number of links in the network. For high-speed networks, the delay may be obtained from (12) by adding the propagation delay τ (Kleinrock, 1992):

$$T = \frac{1}{\gamma}\sum_{i=1}^{m}\frac{f_i}{C_i - f_i} + \tau. \tag{13}$$

In (13), τ is equal to L/c, L denotes the length of the link, and c the speed of light. If L is expressed in kilometers, then: $\tau = \frac{10^{-5}}{3}L$.

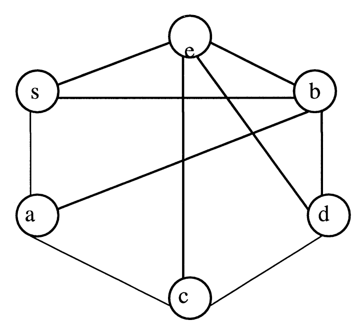
4. Implementation details and results

For the purpose of evaluating the efficiency of the proposed routing method, it has been applied to networks of various sizes. For all the experiments, the average packet size is 1000 bits. The implementation has been realized in *Turbo-Pascal version 7.0 in a* DOS *environment*, on an *IBM PC compatible, Pentium 133 MHz*.

4.1. Routing and flow assignment

A network configuration is represented by a characteristic matrix $M = [M_{ij}]$, $M_{ij} \in \{0, 1\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$, and $\forall(i,j)$ $i \neq j$. In such a matrix, $M_{ij} = 1$ means the link (i,j) exists, $M_{ij} = 0$ means the link (i,j) does not exist. The neural network is represented by a matrix $V = [V_{ij}]$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$, and $\forall(i,j)$ $i \neq j$. In this matrix, each element represents a neuron, and the value V_{ij} represents the output of the neuron (i,j) after its update. The link costs form a matrix having the same dimensions as the matrix of neurons, where each entry represents the length of a link. The traffic matrix denoted by $\Gamma = [\gamma_{ij}]$, with $i, j = 1, 2, \ldots, n$, and $i \neq j$, may be uniform (the same number of packets circulates between each node pair), or random (the number of packets between each node pair is expressed as an integer ranging from 1 to 200). The link capacities (shown in Tables 1 and 2) are selected from options available on the marketplace. Once the network configuration and the associated data are obtained, the routes between node pairs are determined, the flow and capacity are assigned to each link, and the mean delay of the network is finally computed.

For illustrative purposes, the network configuration shown in Fig. 2 has been considered. The application of the neural-network-based routing to this network led to the shortest path between each node pair, as reported in the routing matrix of Table 3. In this table, to each source–destination node pair corresponds a sequence of nodes defining the shortest path found from our routing algorithm. For instance, the shortest path linking the source node s to the destination node b (2nd line of the



Path between the nodes s and d

	s	a	b	c	d	e
s	0	1	0	0	0	0
a	0	0	0	1	0	0
b	0	0	0	0	0	0
c	0	0	0	0	1	0
d	1	0	0	0	0	0
e	0	0	0	0	0	0

Output neuron matrix V_{xi}

Fig. 2. Best path and its output neuron matrix.

Table 1
Options of low and moderate capacities

C (kbps)	9.6	19.2	50.0	100.0	230.40	460.80	921.6	1843.2

Table 2
Options of high capacities

Signal	DS-1	DS-1C	DS-2	2 DS-2	12 DS-1	4 DS-2	DS-3	2 DS-3	4 DS-3	DS-4
C (Mbps)	1.544	3.152	6.312	12.624	18.528	25.248	44.736	89.472	178.94	274.18

Table 3
Routing matrix associated with the network in Fig. 2

Node pair	Sequence of nodes of the shortest path found
s–a	*s–a*
s–b	*s–a–c–d–b*
s–c	*s–a–c*
s–d	*s–a–c–d*
s–e	*s–e*
a–b	*a–c–d–b*
a–c	*a–c*
a–d	*a–c–d*
a–e	*a–s–e*
b–c	*b–d–c*
b–d	*b–d*
b–e	*b–e*
c–d	*c–d*
c–e	*c–d–b–e*
d–e	*d–e*

Table 4
Link flows and capacities of the network in Fig. 2

Link	For a traffic of 10 packets/s		For a traffic of 150 packets/s	
	Flow (kbps)	Capacity (kbps)	Flow (Mbps)	Capacity (Mbps)
s–a	100	230.4	1.50	1.544
s–b	10	19.2	0.30	1.544
s–e	40	50	0.60	1.544
a–b	10	19.2	0.45	1.544
a–c	120	230.4	1.80	3.512
b–d	100	230.4	1.50	1.544
b–e	50	50	0.60	1.544
c–d	140	230.4	2.10	3.512
c–e	10	19.2	0.60	1.544
d–e	40	50	0.30	3.512

table) corresponds to the sequence $s-a-c-d-b$. For a uniform traffic of 10 packets/s and a uniform traffic of 150 packets/s respectively, the resulting link flows and capacities are reported in Table 4. The mean delay T associated with this network configuration is 42 ms for a uniform traffic of 10 packets/sec, and 16 ms for a traffic of 150 packets/s.

On the other hand, by applying the neural-network-based routing to the network configuration of 15 nodes shown in Fig. 3, and for a uniform traffic of 10 packets/s, the mean delay T obtained is 58.2 ms. With a uniform traffic pattern of 150 packets/s between all the node pairs of the network, the mean delay becomes equal to 4.5 ms. The link flows and capacities are reported in Table 5.

4.2. Effect of parameters variations on routing behavior

In this section, the effect of two parameters (the traffic and the size of the network) on the behavior of the proposed routing method is considered. For this purpose, comparisons have been undertaken with other routing methods such as the "shortest distance" and the "minimum number of hops", both methods which are based on the concept of shortest route.

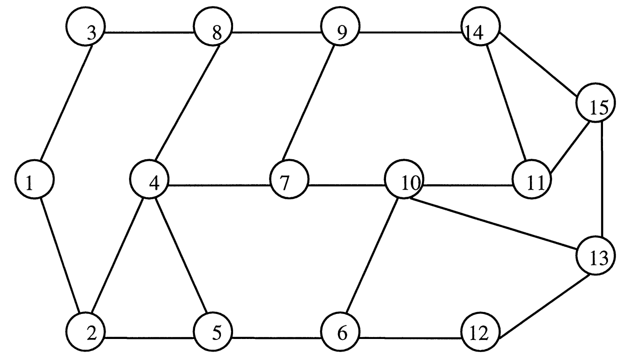


Fig. 3. A network of 15 nodes.

Table 5
Link flows and capacities of the network in Fig. 3

Link	Low traffic		High traffic	
	Flow (kbps)	Capacity (kbps)	Flow (Mbps)	Capacity (Mbps)
1–2	500	921.6	6.15	6.312
1–3	460	460.8	5.85	6.312
2–4	100	230.4	0.75	1.544
2–5	480	921.6	6.00	6.312
3–8	420	460.8	5.40	6.312
4–5	200	230.4	3.30	3.352
4–7	220	230.4	4.20	6.312
4–8	80	100.0	2.40	3.352
5–6	560	927.6	7.20	12.624
6–10	160	230.4	3.30	3.352
6–12	320	460.8	4.50	6.312
7–9	240	460.8	2.40	3.152
7–10	100	230.4	3.00	3.152
8–9	460	921.6	5.40	6.312
9–14	500	921.6	6.60	12.624
10–11	200	230.4	3.00	3.152
10–13	100	230.4	1.80	3.152
11–14	300	460.8	2.40	3.152
11–15	140	230.4	2.40	3.152
12–13	280	460.8	3.90	6.312
13–15	180	230.4	4.20	6.312
14–15	160	230.4	4.80	6.312

4.2.1. Effect of a variation in the traffic level

Consider the network configuration of 10 nodes shown in Fig. 4. The node coordinates are given in Table 6. In order to evaluate the sensitivity of the neural network method relative to the traffic level, the traffic has been varied from 10 to 50 packets/s, always in a uniform fashion. Fig. 5 shows the variations of the mean delay as a function of the traffic level. The mean delays obtained by the neural network method are in general more stable than those resulting from the other two methods.

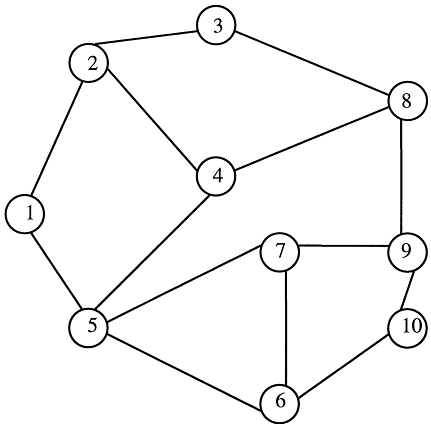


Fig. 4. A second network of 10 nodes.

Table 6
Node coordinates for the network in Fig. 4

Nodes	1	2	3	4	5	6	7	8	9	10
Abscissa	20	20	40	40	30	55	60	70	85	85
Ordinate	60	85	100	70	35	25	60	85	60	30

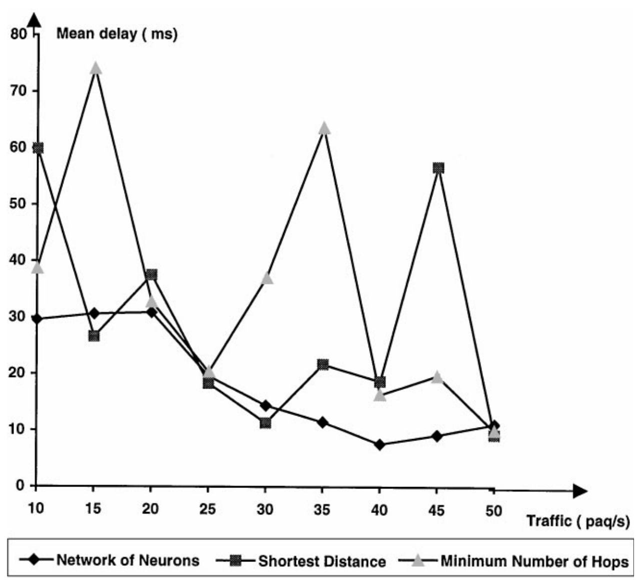


Fig. 5. Variations in the mean delay as a function of the traffic level.

The algorithm of the "shortest distance" essentially corresponds to the Dijkstra's algorithm. The "minimum number of hops" method is an adaptation of the conventional shortest path algorithms, where a unitary length is assigned to each link of the network. Without loss of generality, the paths connecting the node pairs 2 – 10 are selected for this comparative study.

For a uniform traffic of 10 packets/s, an average delay of 59.9 ms is obtained for the shortest distance routing. By this technique, a packet traveling from node 2 to node 10 uses the route 2–1–5–6–10; the related delay is 193.3 ms. The path followed from node 2 to node 10, based on the "minimum number of hops", is always the same (2–4–8–9–10), path which has a length of 4 hops.

4.2.2. *Effect of a variation in the network size*

Consider the networks of 6, 7, 8, 9, 10, 11, and 12 nodes, respectively, which have been generated randomly. Their configuration is presented in Figs. 6–12. For these networks, the traffic is maintained uniformly at 10 packets/s and the results are summarized in Table 7. Fig. 13 shows the variation of the mean delay as a function of the network size expressed in number of nodes and links. According to this figure, there does not exists a meaningful correlation between the mean delay and the size of the network.

4.3. *Comparison with optimal routing algorithms*

For a network of 9 nodes with a uniform traffic of 2 packets/s, the network configuration is shown in

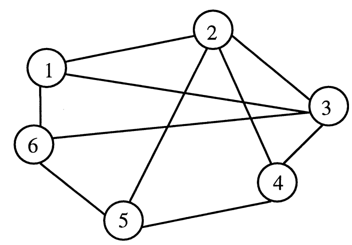


Fig. 6. Network of 6 nodes and 10 links.

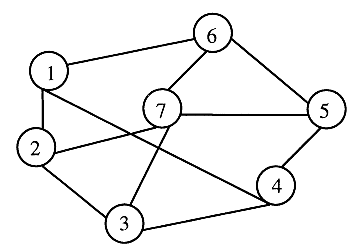


Fig. 7. Network of 7 nodes and 11 links.

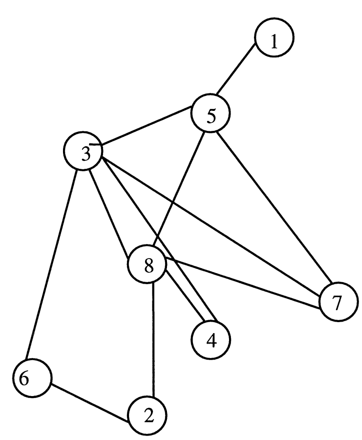


Fig. 8. Network of 8 nodes and 12 links.

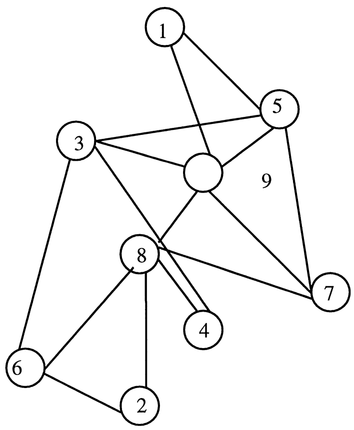


Fig. 9. Network of 9 nodes and 15 links.

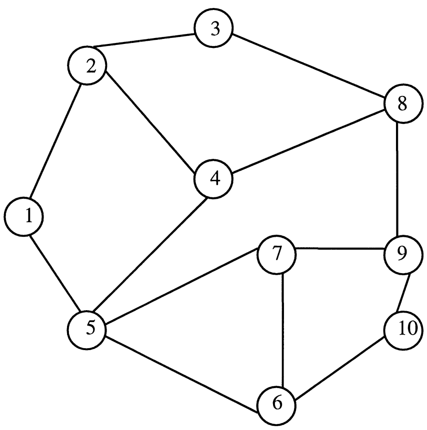


Fig. 10. Network of 10 nodes and 14 links.

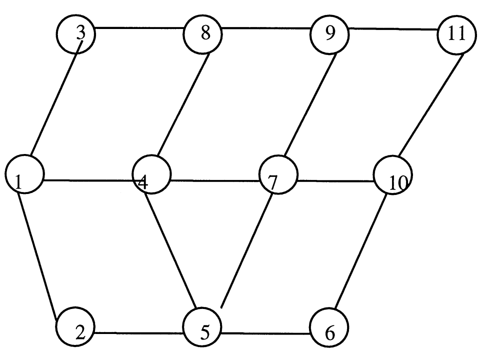


Fig. 11. Network of 11 nodes and 16 links.

Fig. 14. The node coordinates are indicated in Table 8, and the new capacity options are the following: 9.6, 19.2, 28.8, 38.4, 48.0, 57.6, 67.2, 76.8 kbps. The results are reported in Table 9. In spite of the gap that separates the neural network method from those provided by optimal algorithms such as Flow Deviation (FD) (Courtois and Semal, 1980; Fratta and Gerla, 1974), and Bersekas-Gallager (BG) (Bertsekas and Gallager, 1987; Kershenbaum, 1993), this difference is compensated by smaller execution times consumed by the proposed neural-network approach. Obviously, the

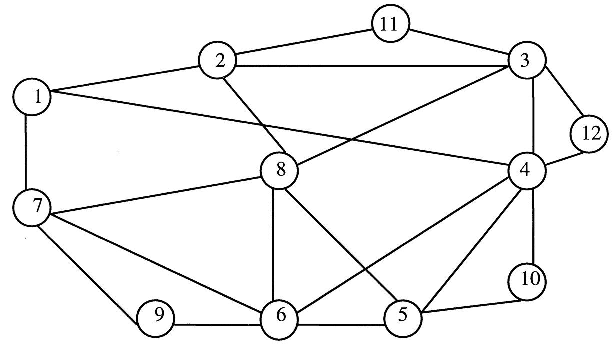


Fig. 12. Network of 12 nodes and 22 links.

Table 7
Comparative results of delay obtained as a function of the network size

Network	Number of nodes	Mean delay (ms)
1	6	46.2
2	7	47.6
3	8	36.8
4	9	34.1
5	10	29.6
6	11	65.9
7	12	72.3

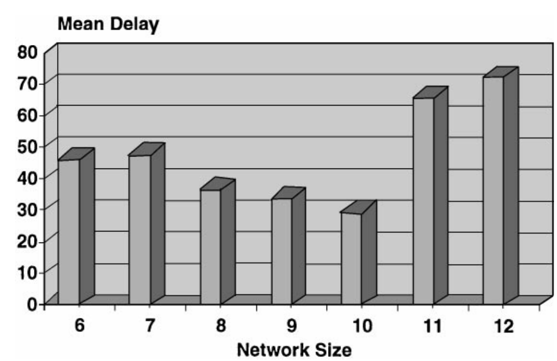


Fig. 13. Variation of delay as a function of the network size.

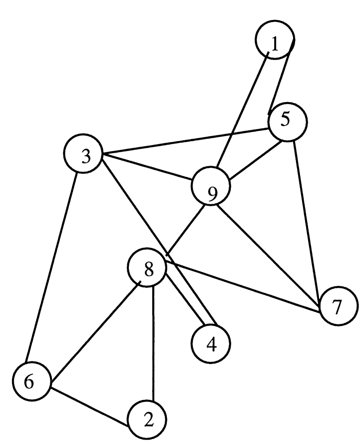


Fig. 14. Network of 9 nodes.

Table 8
Node coordinates of the network in Fig. 14

Nodes	1	2	3	4	5	6	7	8	9
Abscissa	444	60	324	287	433	8	507	176	180
Ordinate	447	16	346	96	396	89	103	191	320

Table 9
Comparison results with optimal routing algorithms

Routing method	T_{mean} (s)	Execution time (s)
Shortest distance	0.795	0.135
Minimum no. of "hops"	0.450	0.130
Neural network	0.33	0.142
Flow deviation	0.176	5.54
Bertsekas–Gallager	0.180	13.94

large number of iterations required by these optimal algorithms before obtaining a solution, as reported in Kershenbaum (1993), causes this contrast in execution times.

5. Conclusion

In this paper, a routing algorithm based on the Hopfield model of artificial neural networks is presented. Routing in a packet switching network consists of determining the best path between each node pair through the network in order to minimize the network delay. The neural-network approach proposed is based on the utilization of an energy function which simulates the objective function used in network optimization.

Computer networks considered in this paper are modeled as non-oriented graphs composed only of full-duplex communication links. The proposed neural-network routing is based on a network representation enabling the designer to match each network configuration with a Hopfield neural network in order to find the best path between any node pair by minimizing the energy function. Some specific parameters must be used to serve as regulators and precision factors, as well as to avoid blocking in a local minimum.

To fix the values of these parameters, Hopfield and Tank (1986) have opted for a formulation which combines mixed legality constraints. Such a choice is simple, but far from efficient since much time is lost avoiding illegal states. Formulating the same problem as an inequation system, Mehmet and Kamoun (1993) found for these parameters some values which potentially led to a great number of iterations before converging.

For experimentation and simulation purposes, only uniform traffic patterns have been considered. (This may be justified by the fact that most of publications use these patterns in order to compare each new routing algorithm with other existing routing algorithms.) The sensitivity of the neural network method relative to the level of traffic has been evaluated. The mean delays obtained by the neural-network approach are generally more stable and less than those determined using conventional routing methods such as the "shortest path" and the "minimum number of hops". Compared with optimal algorithms such as the "Flow deviation" method (Courtois and Semal, 1980; Fratta and Gerla, 1974), and the algorithm of Bertsekas and Gallager (1987) and Kershenbaum (1993), the proposed neural-network method gives delay results slightly less favorable, but in execution times considerably smaller. As a result, the proposed neural-network approach is suitable to be integrated into overall topological design processes, for moderate and high-speed networks subject to quality of service constraints as well as to changes in configuration and link costs.

Acknowledgements

The authors would like to thank the anonymous referees for their insightful comments on the first version of this paper. This work was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada under grant 140264-98.

References

Baransel, C., Dobosiewicz, W., Gburzynski, P., 1995. Routing in multihop packet switching networks: Gb/s challenge. IEEE Network 9 (3), 38–61.

Beaubrun, R., Pierre, S., 1997. A routing algorithm for distributed communication networks. Proceedings of 22nd IEEE Conference on Computer Networks, LCN 97, Minneapolis, Minnesota, USA, November 2–5, pp. 99–105.

Bertsekas, D., Gallager, R., 1987. Data Networks. Prentice-Hall, Englewood Cliffs, NJ, pp. 297–407.

Binh, L.N., Chong, H.C., 1995. A neural network contention controller for packed switching networks. IEEE Transactions on Neural Networks 6 (6), 1402–1410.

Courtois, P.J., Semal, P., 1980. A flow assignment algorithm based on the flow deviation method. Proceedings of ICCC, pp. 77–83.

Dutta, A., Mitra, S., 1993. Integrating heuristic knowledge and optimization models for communication network design. IEEE Transactions on Knowledge and Data Engineering 5 (6), 999–1017.

Drosen, J., 1994. Neural Network Computing. Windcrest/McGraw-Hill, New York.

Fratta, L., Gerla, M., 1974. The flow deviation method: an approach to store-and-forward communication network design. Networks 3, 97–133.

Gavish, B., 1992. Topological design of computer networks — the overall design problem. European Journal of Operational Research 58, 149–172.

Gerla, M., Kleinrock, L., 1977. On the topological design of distributed computer networks. IEEE Transactions on Communications COM 25 (1), 48–60.

Hopfield, J.J., 1982. Neural network and physical systems with emergent collective computional abilities. Proceedings of National Academic Science 79, 2554–2558.

Hopfield, J.J., 1984. Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of National Academy of Science 81, 3088–3092.

Hopfield, J.J., Tank, D.W., 1986. Neural computations of decisions in optimization problems. Biol. Cybern. 52, 141–152.

Jan, R.H., Hwang, F.J., Cheng, S.T., 1993. Topological optimization of a communication network subject to reliability constraints. IEEE Transactions on Reliability 42, 63–70.

Kagiantis, A.G., Papantoni-Kazakos, T., 1997. Operations and learning in neural networks for robust prediction. IEEE Transactions on Systems, Man, and Cybernetics 27 (3), 402–411.

Kamimura, K., Nishino, H., 1991. An efficient method for determining economical configuration of elementary packet-switched networks. IEEE Transactions on Communications 39 (2), 278–288.

Kershenbaum, A., 1993. Telecommunications Network Design Algorithms. McGraw-Hill, New York, p. 368.

Khasnabish, B., 1993. A new method for evaluating packet routing policies in supra-high-speed metropolitan (or wide) area networks. Computer Networks and ISDN Systems 26, 195–216.

Kleinrock, L., 1992. The latency/bandwidth tradeoff in Gigabit networks. IEEE Communications Magazine 30 (4), 36–40.

Lee, S., Chang, S., 1993. Neural network for routing of communication networks with unreliable components. IEEE Transactions on Neural Networks 4 (5), 854–863.

Mehmet, M., Kamoun, F., 1993. Neural networks for shortest path computation and routing in computer networks. IEEE Transactions on Neural Networks 4 (6), 94–954.

Neumann, I., 1992. A system for priority routing and capacity assignment in packet switched networks. Annals of Operations Research 36, 225–246.

Pierre, S., Beaubrun, R., 2000. Integrating routing and survivability in fault-tolerant computer network design. Computer Communications 23 (4), 317–327.

Rauch, H.E., Winarske, T., 1988. Neural networks for routing communication traffic. IEEE Control System Magazine 8 (2), 26–30.

Rumelhart, D., Widrow, B., Lehr, M., 1984. The basic idea in neural networks. Commications of ACM 37 (3), 87–92.