2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

A packet routing method for a random network by a stochastic neural network

Takayuki Kimura[†,‡], Hiroyuki Nakajima[‡] and Tohru Ikeguchi[†]

†Graduate School of Science and Engineering, Saitama University
255 Shimo-Ohkubo, Sakura-ku, Saitama 338-8570, Japan
‡School of Engineering, Kinki University
1 Takaya-Umenobe, Higashi-Hiroshima 739-2116, Japan
Email: kimura@nls.ics.saitama-u.ac.jp

Abstract—We propose a new algorithm for controling packet routing using a neural network. First, we show that the conventional method for a packet routing control using a neural network is not so effective when it is applied to a network with an irregular topology. To overcome this problem, we modified the method with stochastic effects. We confirmed that our algorithm is very effective not only for regular but also for irregular networks.

1. Introduction

Recent years, amount of data flow in computer network are rapidly increased because of the growth of users and information. In computer networks, many packets of various sizes are exchanged, however, they are often deleted and delayed because of nonstationarity of the network. Thus, it is important to control the computer networks more effectively.

A packet routing has generally two strategies; a centralized control and a decentralized control. The centralized control is a strategy that a centralized unit controls all packets routing in the network. The centralized control shows a good performance in a small-scale network. However, in a large-scale network, the centralized control does not work well. If the network size becomes large, the central unit has huge computational load. Therefore, it is impossible to control the packets to send their destinations realistically using the centralized control. On the other hand, the decentralized control is more applicable than the centralized control for the large-scale network because each unit sends packet autonomously and adaptively.

Various models using the decentralized control have been proposed. Recently, a new method of packet routing was proposed by Horiguchi and Ishioka using a neural network[1] (the NN method). Its effectiveness has been evaluated on an average number of packets arrived at the objective node and discarded at their objective node. The NN method shows good performance for a network with a regular topology. However, effectives of the NN method for an irregular topological network is not yet clarified. In general, real networks have an irregular topology. Then, we first confirmed its effectiveness for an irregular topological network by comparing with Dijkstra algorithm[2]. As a result, the NN method shows poor performance because of nonuniform distributions of packets. To overcome this problem, we proposed a modified model of the NN method which introduces stochastic effects. We evaluated if effectiveness by computer simulations.

2. Model of a computer network

The computer network consists of nodes and links (or connections). Each packet is sent from a node to a node through links. A packet can be sent at the nodes and multiple packets can be received simultaneously. Every node has a buffer which stores some of packets and every packet is sent according to First In First Out control. The packet sent to a node is deleted when the buffer of the node is full. Moreover, every packet has an upper limit of packet movement. Thus, every packet is deleted if it exceeds this limit.

3. Link selection

In the decentralized control, each node selects the optimum adjacent node autonomously to send the packet to the objective node. To achieve this optimum selection, each node uses, for example, the distance from node to the objective node or amount of packets in the adjacent nodes. We call this selection a link selection. In other words, the packet routing by the decentralized control is an integral of the link selections.

4. Routing method using a neural network

Horiguchi and Ishioka proposed a method for a packet roting that employs neural networks on link selections[1]. In this method, a network has N nodes and the i-th node had N_i adjacent nodes. Then, N_i neurons are assigned to each node, and the il-th neuron corresponds to the connection between adjacent nodes ($l = 1, \ldots, N_i$). The neurons are fully connected each other. Moreover, the variable v_{il} ($0 \leq v_{il} \leq 1$) is designated as an output of the il-th neuron, the il-th neuron is adjacent to the i-th node. If $v_{il} = 1$ (the il-th neuron fires), the packet at the node i is sent to the node l.

To minimize the number of packets stayed at each node, to reduce the distance to the objective node of a packet and to make only one neuron fire at each node, we defined the following function E:

$$
\begin{aligned}
E = & \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N_i}\sum_{l=1}^{N_i} J_{ik,il} v_{ik} v_{il} \\
& - \eta \sum_{i=1}^{N}\sum_{l=1}^{N_i}\left\{1 - \frac{1}{b_l}\left(q_l + \frac{1}{2}\sum_{j\neq i}^{N_l} v_{jl}\right)\right\} v_{il} \\
& - (1-\eta)\sum_{i=1}^{N}\sum_{l=1}^{N_i}\left(1 - \frac{2d_l}{d_c}\right) v_{il} + \xi \sum_{i=1}^{N}\left(\sum_{l=1}^{N_i} v_{il} - 1\right)^2 . \quad (1)
\end{aligned}
$$

where $J_{ik,il}$ is a connection weight between the ik-th neuron and the il-th neuron; η is a control parameter which changes priority of the second and third term; b_l is a buffer size of node l; q_l is a number of packets in the buffer of node l; d_l is the shortest distance from node l to an objective node; d_c is a control parameter which expressed the size of the computer network; and ξ is a control parameter which guarantees uniqueness of link selection.

In Eq.(1), the first term expresses inhibition of sending packets. The second term expresses the loads of the adjacent nodes. The third term expresses the distance from node i to the objective node of a ready-to-go-out packet at node i. The last term expressed a single neuron firing at each node.

We assume that node l is an optimum node in Eq.(1). In other words, node l is closest to the objective node from node i and node l has a small number of packets. In Eq.(1), E decreases if and only if v_{il} gets close to 1. Therefore, every node selects a link optimally if the state of the neural network changes as the energy function E decreases.

We changed the internal state of the neural network in a similar way as Hopfield and Tank[3], then the output of a neuron v_{il} is written in the following. We first introduced a new variable h_{il} which expresses the state of the neuron defined as follows.

$$
v_{il} = \begin{cases} \dfrac{1}{1+\exp(-\beta h_{il})} & \left(\begin{array}{l}\text{if the nodes } i \text{ and } l \text{ are} \\ \text{adjacent}\end{array}\right) \\ 0 & \text{(otherwise))} \end{cases} \quad (2)
$$

The variable of h_{il} is changed when node i is adjacent to node l.

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} h_{il}(t) = & -\sum_{k=1}^{N_i} J_{ik,il} v_{ik}(t) \\
& + \eta\left\{1 - \frac{1}{b_l}\left(q_l + \sum_{j\neq i}^{N_l} v_{jl}(t)\right)\right\} \\
& + (1-\eta)\left(1 - \frac{2d_l}{d_c}\right) - 2\xi\left(\sum_{k=1}^{N_i} v_{ik}(t) - 1\right). \quad (3)
\end{aligned}
$$

We obtain Eq.(3) by differentiating E partially with respect to v_{il}. Then, Eq.(3) is equivalent to $\dfrac{\partial E}{\partial v_{il}}$. Also, from Eq.(2), we obtain,

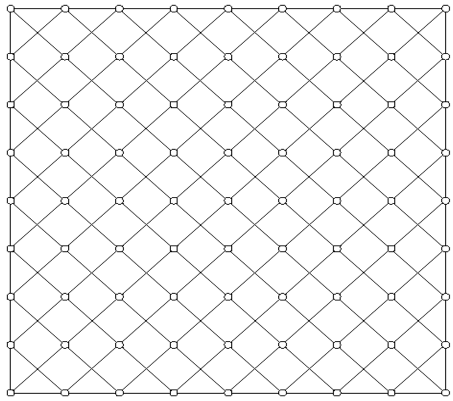

Figure 1: A regular network.

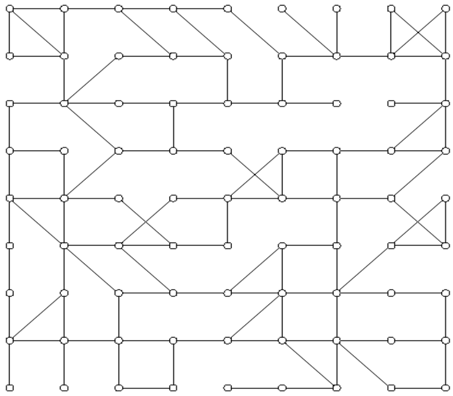

Figure 2: A irregular network.

$$
\frac{\mathrm{d}v_{il}}{\mathrm{d}h_{il}} = \frac{\beta}{4\cosh^2\left(\frac{1}{2}\beta h_{il}\right)} \quad (4)
$$

Thus,

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}E = & \sum_{i,l=1}^{N} \frac{\partial E}{\partial v_{il}} \frac{\mathrm{d}v_{il}}{\mathrm{d}h_{il}} \frac{\mathrm{d}h_{il}(t)}{\mathrm{d}t} \\
= & -\sum_{i,l=1}^{N}\left(\frac{\partial E}{\partial v_{il}}\right)^2 \frac{\beta}{4\cosh^2\left(\frac{1}{2}\beta h_{il}\right)} \leq 0 \quad (5)
\end{aligned}
$$

Eq.(5) indicates that the energy function E decreases when h_{il} changes according to Eq.(3). The energy function Eq.(1) decreases by the method of steepest descent.

5. The problems of the NN method and its modification

Performance of the NN method, as described in section 4, is examined with a regular network in which all nodes are allocated in a rectangular pattern by computer simulations[1]. However, a real network has an irregular topology. Thus, it is significant to verify the effectiveness of the NN method for the real network. We are also interested in how the performance would be if the topology of the network changes from a regular to a random one. Moreover, the basic strategy of the NN method[1] is based on link selections optimally by balancing the shortest distance of an objective node and congestion of adjacent nodes.

To set the lengths of a link between adjacent nodes to the quantity of packets in adjacent node, we also proposed the routing method using Dijkstra algorithm[2], which is one of the standard algorithm for solving the shortest path problem.

In this paper, we first confirmed the efficiency of the NN method for some different networks, one is a regular and another is an irregular network, comparing with the Dijkstra method and the NN method.

The regular and irregular networks are shown in Figs.1 and 2. The network shown in Fig.1 is used in Ref.[1]. We conducted computer simulations of packet routing in the following procedures.

First, we generated packets randomly at all nodes in the network using uniformly distributed random numbers; each packet has an objective node and the objective nodes assigned randomly using uniformly distributed random numbers. Next, the link selection is simultaneously conducted at every node and a packet is sent to the adjacent node. A packet sent from the node was stored at the tail of the buffer of the adjacent node. We set a buffer size $b_l = 200$ and a constraint for packet movement to 20. A packet is deleted when the packet exceeds these constraints.

We repeated this motion link selection and packet sending for 10,000 iterations. We fixed the total number of packets in the network. We added a new packet when a packet was deleted. A new packet is generated at a node and its objective node is randomly decided using uniformly random numbers.

We explain the link selection of the NN method. First, the variable h_{il} is converged to a steady value by solving Eq.(3) using Euler's method. In the Euler's method, we set the number of iterations 50 with step size 1. Then, v_{il} is computed by Eq.(2) and a packet in node i is sent to node l if v_{il} is larger than the threshold θ; we choose $\theta = 0.9$. We set $\beta = 3.0$ and $\eta = 0.2, 0.6, 0.8, 0.9$ and $\xi = 0.3$. We also set d_c as an average longest path length.

The Dijkstra algorithm searches the shortest path from a node to the other nodes in a nonnegative directed graph[2]. In this paper, we set the lengths of links connecting to the adjacent node the amount of packets in the adjacent node. Moreover, we set the lengths of links from adjacent nodes to other node to 1. We updated the lengths of links at each iteration and searched the shortest paths from the node to the other nodes using Dijkstra algorithm because the lengths of links to adjacent node is set by amount of packets in its node. Then, a packet is sent to the adjacent node which is on the shortest path.

We described an average number of packets at each node by N_p, and an average number of packets arrived at their objective node by A:

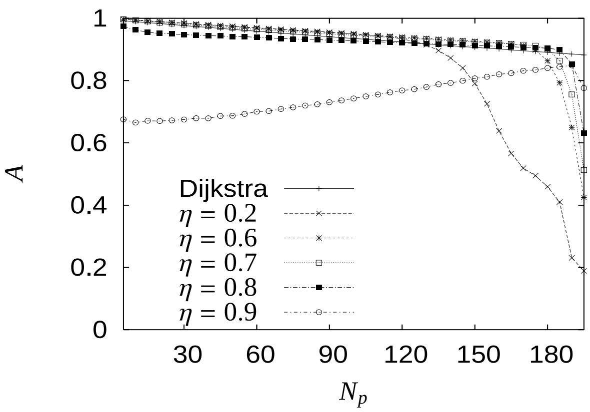

Figure 3: An average number of packets arrived at their objective node in the regular network.

$$A = \frac{N_a}{N_g}, \tag{6}$$

$$D = \frac{N_{dp}}{N_g}, \tag{7}$$

where N_{tp} is the total number of packets in the network at a certain moment; N_a is the number of packets which have arrived at their objective node; N_g is the number of all packets generated in the network.

Results for the regular and irregular networks are shown in Fig.3 and 4, respectively. In the case of the regular network, the NN method (except cases of $\eta = 0.2$ and $\eta = 0.9$) and the Dijkstra method exhibits high average number of arrived packets, even N_p is large. The NN method with $\eta = 0.8$ shows the most optimum solution in the regular network (Fig.3). However, for the irregular network, the performance of the NN method exhibits worse than the Dijkstra model. In addition, the value of A rapidly decreased when N_p is high (Fig.4). These results indicates that the NN method has poor performance if the network topology is irregular.

6. Proposed method

In previous section, we clarified that the NN method has poor performance for the irregular network. We consider the reason as follows. In the irregular network, packets are sent intensively to a particular node frequently. If the number of packets exceeds the size of buffer the node, the overflowed packets are deleted. Moreover, in the irregular network, some nodes sometimes make a loop. If a packet is trapped to the loop, escape from the loop is difficult because there is no optimum adjacent node. So, the packet will be dead.

To avoid the death of the packet, we modified the NN method. In the conventional method, an optimum adjacent nod, to which the packet is sent, is determined by a deterministic rule. We introduced a stochastic effect to the

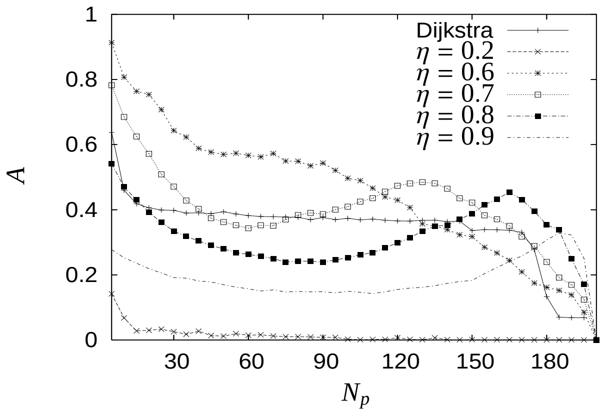

Figure 4: An average number of packets arrived at their objective node in the irregular network.

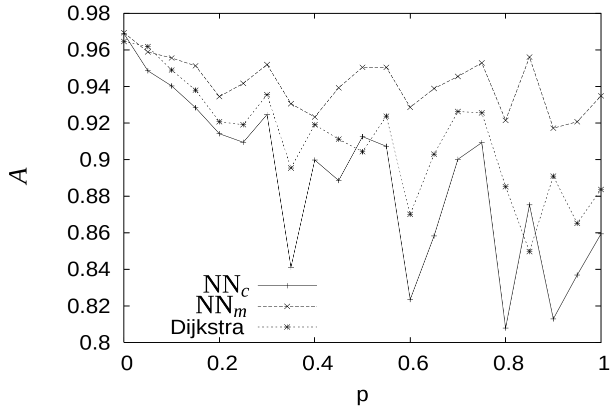

Figure 5: An average number of packets arrived at their objective node in the randomized networks.

determination. We changed the variable η in Eq.(1) randomly using uniformly distributed random numbers. This modification implies that the priority between the shortest distance of the packets to the objective node and decentralization of loads of the adjacent nodes is changed randomly when the node selects the adjacent node. Therefore, it is expected to avoid the packets sent to a particular node. We call the conventional method NN_c and the modified one NN_m

We evaluate the NN_m, the NN_c and the Dijkstra algorithm for the irregular network (the network structure is shown in Fig.2). Moreover, we also evaluated these methods for randomized networks. The randomized networks are generated in a similar way as Watts and Strogatz[4]. Starting from the regular network as shown Fig.1, we rewired each link at random with a probability p ($0 \leq p \leq 1$). We also introduced a constraint that each link cannot be connected to a further node beyond three links. This construction allows us to tune the network between regular ($p = 0$) and disorder ($p = 1$).

We set the variable of $\eta = 0.8$ in Eq.(3) for the NN_c and we change η for the NN_m randomly between 0.1 and 0.9. In these simulations, we used the same experimental assumption in section 5. We evaluated these methods by an average number of packets arrived at the objective node (A) for the irregular network as shown in Fig.5. The results indicates that the NN_m exhibits high performance when N_p increases. The NN_m is the best for every N_p.

We also show the average number of packets of NN_c, NN_m and the Dijkstra methods for the randomized networks in Fig.6. In this simulation, we set $N_p = 50$ for every p. In Fig.6, the three models show high performance when p is small. However, the performance of the Dijkstra algorithm and NN_c becomes low when p is large. On the other hand, the performance of the NN_m keeps high performance.

From these results, the NN_m has higher average number of packets arrived at the objective node for the regular and irregular network.

7. Conclusions

In this paper, we proposed a modified method for packet routing by a neural network. We introduced the stochastic effects; a random selection for the priority between the shortest distance of the packet to the objective node and decentralization of loads of the adjacent nodes. By the stochastic effects, the proposed method shows high performance comparing to the conventional method when the network structure changes from regular to irregular one. The research of TI is partially supported by Grant-in-Aid for Scientific Research (B) from JSPS (No.16300072)

References

[1] T.Horiguchi and S.Ishioka, “Routing control of packet flow using neural network,” *Physica A*, vol.297, pp.521–531, 2001.

[2] D.Bertsekas and R.Gallager, “Data Networks,” *Prenticehall.*, 1987.

[3] J.J.Hopfield and D.W.Tank, “Neural computation of decisions in optimization ploblems,” *Biological Cybernetics.*, vol.52, pp.141–152, 1985.

[4] D.J.Watts and S.H.Strogatz, “Collective dynamics of small-world networks,” *Nature.*, vol.393, pp.440–442, 1998.