

Neural-Based Routing Method for Alleviating Congestion in Complex Networks

Takayuki Kimura¹, Tohru Takamizawa², Takafumi Matsuura¹

¹Department of Electrical and Electronic Engineering, Nippon Institute of Technology, Saitama, Japan

²Graduate School of Electronics, Information and Media Engineering, Nippon Institute of Technology, Saitama, Japan

Email: tkimura@nit.ac.jp, e1102304@estu.nit.ac.jp, matsuura@nit.ac.jp

Received 18 June 2016; accepted 18 July 2016; published 21 July 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we propose a new packet routing strategy that incorporates memory information for reducing congestion in communication networks. First, we study the conventional routing strategy which selects the paths for transmitting packets to destinations using the distance information and the dynamical information such as the number of accumulating packets at adjacent nodes. Then, we evaluate the effectiveness of this routing strategy for the scale-free networks. From results of numerical simulations, we conclude that this routing strategy is not effective when the density of the packets increases due to the impermeability of the communication network. To avoid this undesirable problem, we incorporate memory information to the routing strategy. By using memory information effectively, packets are spread into the communication networks, achieving a higher performance than conventional routing strategies for various network topologies, such as scale-free networks, small-world networks, and scale-free networks with community structures.

Keywords

Congestion Control, Packet Routing Problems, Decentralized Control, Complex Networks

1. Introduction

The transmission rate of data packets in communication networks is drastically increasing due to the widespread use of mobile devices. If the number of packets increases sufficiently, then packet congestion occurs in communication networks. Transmissions of packets to their destinations are then blocked, and in the worst case,

the packets may be removed from the network by a buffer constraint or a time-to-live constraint. Therefore, the study and analysis of packet congestion has been given significant attention in recent years [1]-[17].

The shortest path algorithms adopted in real-world systems, such as the Dijkstra [18], Bellman-Ford [19], and Warshall-Floyd [20] algorithms, face serious challenges if the number of data packets continues to increase [15]. Specifically, these shortest path algorithms transmit packets using only information regarding distances in the communication network, and so routers that lie on a large number of shortest paths easily become congested. Thus, it is imperative that the transmission algorithm is enhanced to ensure reliable communication through the network.

Understanding the dynamics of packet transmission on networks is also an important task in analyzing the onset of traffic congestion. From this viewpoint, Ohira *et al.* [17] illustrated the onset of phase transition behavior from a free-flow state to a congested state using square lattice-type network models. Arenas *et al.* [16] analyzed this phase transition behavior using hierarchical branching networks. Zhao *et al.* [13] showed that the heterogeneous property of a network inhibits traffic congestion in the case that the transmitting performance of each node is proportional to its degree.

On the other hand, the development of efficient routing methods that drastically reduce the congestion of a network presents another important task for carrying a large volume of data traffic. Recent studies concerning the design of routing strategies have evolved according to two basic ideas. The first is to select paths for transmitting packets based on only local information of the network, such as degree information or the number of packets waiting at adjacent nodes [4] [6] [8] [10] [11]. Such routing strategies can be applied to networks of any size because it is not necessary to collect distance information for the whole network. The second idea is to utilize global information, such as shortest distance information for the network. For example, Echenique *et al.* [14] [15] proposed a deterministic routing strategy, which calculates efficient paths for the sending of packets by using distance information and dynamic information. Yan *et al.* [9] proposed a routing strategy that uses both distance and degree information for the network. Zhang *et al.* [7] proposed an adaptive routing strategy based on measuring the accumulated number of packets on shortest paths. A routing strategy incorporating traffic awareness was proposed by Wang *et al.* [5]. Further, as an example of using global information, a routing method has been proposed based on mutually connected neural networks (MCNN routing) [21]. This can be further enhanced by using reinforcement learning [22] or stochastic effects [23]. The MCNN routing method searches all nodes adjacent to the sending node in order to route packets using the energy minimization principle, based on mutually connected neural networks. It is well known that mutually connected neural networks give an effective method for solving hard classes of problems, namely NP-hard problems. Furthermore, MCNN routing achieves good arrival rate for packets in regular topological communication networks. However, it performs poorly for irregular topological communication networks, because of the nonuniform distribution of packets [23]. In order to improve its performance in this case, we have previously proposed a neural-based routing strategy that uses transmitting information [24]. Concerning this strategy, it still remains to perform a performance evaluation regarding the alleviation of packet congestion in order to clarify how many packets the routing strategy can transmit in the case that the number of flowing packets increases. Such an evaluation is necessary if the routing strategy is to be employed in real communication networks. In addition, the congestion rate [16] can be used to determine the capability of avoiding packet congestion in a communication network model in which the number of packets to be transmitted continues to increase. This has been adopted in recent studies regarding the development of routing strategies [4] [6] [8] [10] [11].

From the above viewpoints, our neural-based routing method [24] is comprehensively evaluated in this paper using various performance measures such as the packet congestion rate, congestion levels, average arrival rate of packets, and average number of lost packets. The results show that the proposed neural-base routing method has a strong ability to carry large volumes of data compared with the conventional neural-based routing methods, and utilizes the transmitting information effectively.

This paper is organized as follows. In Section 2, we first describe how the routing method is constructed based on mutually connected neural networks, and then evaluate the MCNN routing method using the packet congestion rate. The modified version that utilizes transmitting information is then described in Section 3. Furthermore, our proposed neural-based routing method is evaluated in comparison to other conventional neural-based routing methods in Section 4. Finally, we conclude this paper in Section 5.

2. Routing Method Based on Mutually Connected Neural Networks

First, we explain how to construct the original MCNN routing method [21] because our proposed method is

based on this. To proceed, we first describe the communication network model. Here, the communication network model is composed of a set of nodes and a set of links. Each node represents a host or a router, and each link represents a physical connection between two nodes. The node has a buffer to store packets, and can receive multiple packets simultaneously from its adjacent or connected nodes. The packets, whose sources and destinations are randomly determined, are then generated in the communication network. A packet is transmitted to a node according to the FIFO (first-in first-out) principle and is removed from the communication network if it arrives at its destination.

The communication network model has N nodes, where the i th node is connected to N_i adjacent nodes. Next, we assign a mutually connected neural network to each node. In this MCNN routing method, each neuron in the neural network corresponds to a connection between the nodes. For example, the j th neuron, which is the j th neuron in the i th neural network, represents the connection between the i th node and the j th node. Hence, a mutually connected neural network with N_i neurons is assigned to each node. **Figure 1** illustrates an example of the neural coding for the communication network model.

In the MCNN routing method, each node determines one of the adjacent nodes as the transmitting node at each iteration by minimizing the following energy function:

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

where $J_{ik,il}$ is the connection weight between the ik th neuron and the il th neuron, b_l is the buffer size of the l th adjacent node, q_l is the number of packets stored in the l th adjacent node, d_l is the shortest distance from the l th adjacent node to the destination of a packet at the i th node, d_c is the longest distance among all shortest paths, η is the control parameter that decides the priorities of the first and second terms, and ξ is the control parameter that guarantees the uniqueness of the firing neurons.

In Equation (1), the second term expresses the load distribution of adjacent nodes, and the third term is the distance for a packet at the i th node to the destination. The last term guarantees that only a single neuron fires in each neural network.

The internal state of the mutually connected neural network is updated using the method proposed by Hopfield and Tank [25]. The output of the il th neuron v_{il} is described as follows:

$$v_{il} = \begin{cases} \frac{1}{1 + \exp(-\beta h_{il})} & \text{(if the } i\text{th node is adjacent to the } l\text{th node),} \\ 0 & \text{(otherwise).} \end{cases} \tag{2}$$

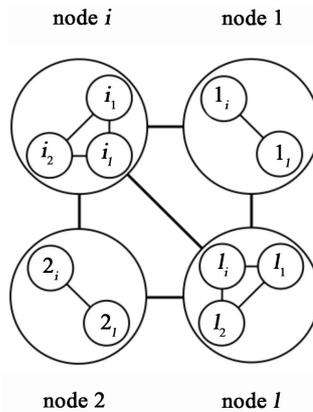


Figure 1. Schematic representation of the neural network coding in this paper. In this example, the i th node has three adjacent nodes. The neural network of the i th node is therefore constructed from three neurons that correspond to adjacent nodes 1, 2, and l .

Moreover, when the i th and l th nodes are adjacent, the internal state of the il th neuron h_{il} changes according to the following differential equation:

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

Note that Equation (3) is obtained by partially differentiating Equation (1) with respect to v_{il} . In other words, Equation (3) is equivalent to $-\frac{\partial E}{\partial v_{il}}$.

From Equation (2) we obtain that

$$\frac{dv_{il}}{dh_{il}} = \frac{\beta}{4 \cosh^2 \left(\frac{1}{2} \beta h_{il} \right)}. \quad (4)$$

Thus,

$$\frac{d}{dt} E = \sum_{i,l=1}^N \frac{\partial E}{\partial v_{il}} \frac{dv_{il}}{dh_{il}} \frac{dh_{il}}{dt} = -\sum_{i,l=1}^N \left(\frac{\partial E}{\partial v_{il}} \right)^2 \frac{\beta}{4 \cosh^2 \left(\frac{1}{2} \beta h_{il} \right)} \leq 0. \quad (5)$$

Equation (5) indicates that the energy function E always decreases when h_{il} changes according to Equation (3). In other words, the energy function of Equation (1) decreases according to the steepest descent method.

In this method, the firing of the neuron is determined using the threshold θ . That is, the il th neuron fires if $v_{il} \geq \theta$ and does not if $v_{il} < \theta$. When the il th neuron fires, a packet at the buffer head of the i th node is transmitted to the l th adjacent node.

The l th adjacent node of the i th node is selected as the transmitting node if it has the shortest distance to the destination of the packet and a smaller number of packets in its buffer among all of adjacent nodes.

As mentioned above, η in Equation (1) determines the priority of the load distribution of the adjacent nodes and the distance of packet to the destination. If $\eta = 1$, then the transmitting packets are distributed widely throughout the network, in order to reduce heavily congestion at a node. However, the transmissions of almost all packets to their destinations will be delayed because in this case, the distance information for packets is ignored. On the other hand, if $\eta = 0$, the packets are transmitted to their destinations along the paths with the shortest distance, with the result that many packets become trapped at nodes with large degree.

Next, we evaluate the effectiveness of the MCNN routing method. Because real communication networks have the scale-free property [26], we adopt the scale-free topology here as the computer network topology. We construct scale-free networks by the following procedure [26]. We begin with a complete graph with m_0 nodes. Then, we insert a new node with m_0 links at every time step. Next, the links of the newly added node are connected to the existing nodes in the network with probability $\Pi(k_i) = k_i / \sum_{j=1}^N k_j$, where k_i is the degree of the i th node ($i = 1, \dots, N$), and N is the number of the nodes at the current iteration. For this study, we fixed $m_0 = 2$.

We conducted numerical experiments according to the following procedure. We generated p packets whose sources and the destinations are randomly determined at each iteration. Then, link selection is simultaneously conducted at every node, and the packets that are ready to exit each node are transmitted to adjacent nodes. A packet is removed from the network when it reaches its destination. In addition, the transmission of a packet is canceled if the buffer of the node to which the packet should be transmitted is full.

We set the number of iterations, T , to 3000, and the buffer size of each node, B , to 100. We set $\beta = 3.0$, $\eta = 0.1, 0.3, 0.5, 0.7$, and 0.9 , and $\xi = 0.3$ in Equation (3), and set the threshold for the state of a neuron as $\theta = 0.9$. We conducted 10 simulations for five different values of η , and averaged the results.

We adopted the packet congestion rate [16] to evaluate the MCNN routing method. The packet congestion rate defined as follows:

$$C = \frac{1}{T} \sum_{t=1}^T c(t). \quad (6)$$

In Equation (6), C is the packet congestion rate, T is the number of iterations, and $c(t)$ is defined by

$$c(t) = \begin{cases} \frac{n(t) - n(t-1)}{p} & (n(t-1) < (N(B-p))), \\ 1 & (\text{otherwise}). \end{cases} \quad (7)$$

In Equation (7), $n(t)$ is the number of packets at the t th iteration, N is the number of nodes, p is the number of packets generated at each iteration, and B is the buffer size of every node.

Figure 2 illustrates the packet congestion rate for the MCNN routing method. It can be seen that the packet congestion rate increases more quickly in the cases that $\eta = 0.1$ and 0.9 than for any others, because of the unbalanced setting between the distance to the destination and the distribution of the packets defined by Equation (1). On the other hand, with $\eta = 0.5$ this method avoids packet congestion, even if the number of generated packets increases. Hereafter, we set $\eta = 0.5$ for the MCNN routing method.

To analyze why the MCNN routing method displays a poor congestion rate if the number of flowing packets increases, we next measure the congestion level of nodes in the computer network. To evaluate the congestion levels of the nodes, we employ a computer network model in which a fixed number of packets are flowing [23] [24] [28]. As in the references [23] [24] [28], we introduce a buffer constraint and a packet moving constraint. Packets are then removed from the communication network if they are transmitted to nodes that have full buffers. In addition, packets are also removed when they exceed a fixed threshold for packet movements. If a packet arrives at its destination or is removed from the network, then we add an additional packet, to preserve the number of the packets flowing in the network.

In these simulations, we set the buffer size of every node to 100 and limited the number of packet movements between nodes to 20. The congestion level of the i th node at the t th iteration, $0 \leq L_i(t) \leq 1$, is defined as follows:

$$L_i(t) = q_i(t)/B. \quad (8)$$

In Equation (8), $q_i(t)$ is the number of queued packets of the i th node at the t th iteration, and B is the buffer size of every node. If $L_i(t)$ is close to 1, this indicates that the i th node has many packets waiting for further transmission. That is, the i th node is heavily congested. On the other hand, the i th node can afford to transmit packets to the next node quickly if $L_i(t)$ is close to 0.

Figure 3 illustrates the congestion levels of the nodes for the MCNN routing method. In **Figure 3**, the congestion levels of the hub nodes with large degree clearly increase as the number of iterations increases. This indicates that the packets gradually accumulate at the hub nodes in the MCNN routing method.

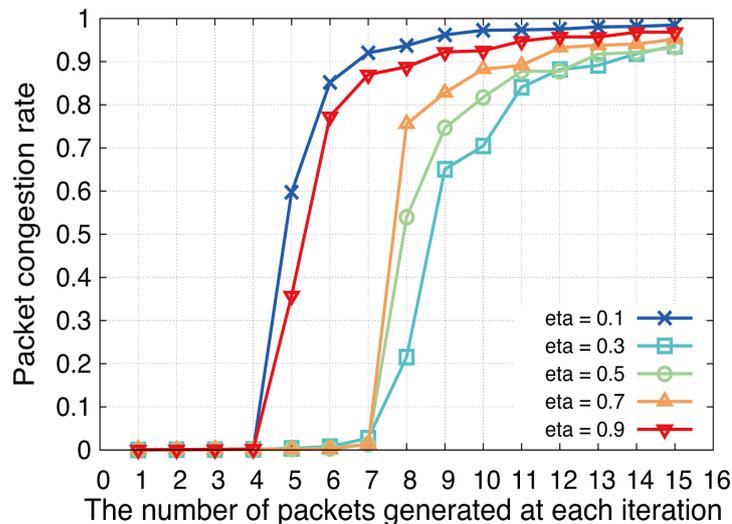


Figure 2. Packet congestion rate for MCNN routing method.

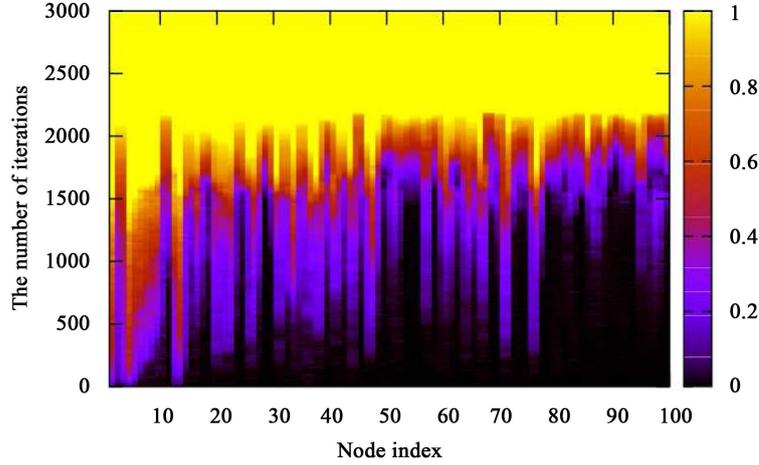


Figure 3. Congestion levels of nodes for the MCNN routing method. In this figure, we fixed the number of flowing packets to 10. The node index is sorted in descending order of the degrees of the nodes.

3. Neural-Based Routing Method Using Transmitting Information

We have clarified that the conventional neural-based routing method [21] leads to congestion, because congestion at the hub nodes cannot be prevented. The reason why packet congestion increases for the conventional neural-based routing method may relate to the convergence to local minima of Equation (1) because only the shortest distance between a packet and its destination and the number of packets queued at adjacent nodes are considered while choosing an adjacent node for the routing of the packet.

One promising approach to overcome these problems is to consider all of the queuing information for every node. However, this requires heavy processing costs because such information constantly changes depending on the transmitting states of the packets. Another effective way might be to utilize limited queuing information for the nodes that lie on the shortest distance path between the packet and its destination. Hereafter, we refer to this limited queuing information as transmitting information. From this viewpoint, we propose a neural-based routing method that uses transmitting information by modifying the original energy function defined by Equation (1) as follows:

$$E = -\eta \sum_{i=1}^N \sum_{l=1}^{N_i} \left(1 - \frac{\sum_{m=1}^{\zeta} \omega^{m-1} \left(\frac{q_{\chi_{lm}}}{b_{\chi_{lm}}} \right)}{\zeta} \right) v_{il} - (1-\eta) \sum_{i=1}^N \sum_{l=1}^{N_i} \left(1 - \frac{d_l}{d_c} \right) v_{il} + \xi \sum_{i=1}^N \left(\sum_{l=1}^{N_i} v_{il} - 1 \right)^2. \quad (9)$$

Here, we define the subset of nodes $R_l(t) \subset V$, whose elements are the nodes that lie on the shortest distance path to the destination passing through the l th adjacent node from the i th node at the t th iteration. That is,

$$R_l(t) = \{ \chi_{l1}(t), \chi_{l2}(t), \chi_{l3}(t), \dots, \chi_{l\zeta}(t) \}$$

where $\chi_{lj}(t)$ is the index of the j th node on the shortest distance path from the l th node at the t th iteration and ζ is the acquisition range of the transmitting information. Because the shortest distance path varies according to the destination node for each packet, the elements of $R_l(t)$ also change at each iteration.

In Equation (9), $b_{\chi_{lm}}$ is the buffer size of the node χ_{lm} , $q_{\chi_{lm}}$ is the number of packets queued at the node χ_{lm} , d_l is the shortest distance from the l th adjacent node to the destination node of a packet at the i th node, d_c is the longest path length in the network, η is the control parameter that determines the priorities of the first and second terms, ξ is the control parameter that guarantees the uniqueness of firing neurons, ζ is the acquisition range of the transmitting information for transmission, and $\omega(0 < \omega < 1)$ is a decaying parameter for the transmitting information.

In Equation (9), the first term expresses the transmitting information of nodes on the shortest distance path

through the l th adjacent node, and the second term is the distance of a packet from the l th adjacent node to the destination. The third term guarantees that only one neuron fires in the neural network. If the distance to the destination of a packet from the l th adjacent node is the shortest among all of the adjacent nodes, and the transmitting information of the shortest distance path to the destination starting through the l th adjacent node is the lowest among all of the paths, then the l th adjacent node is selected as the transmitting node.

The cost of calculation for the conventional neural-based routing method [21] becomes $O(N^3)$ because it calculates the packet transmitting information between the j th and l th adjacent nodes as expressed by v_{jl} in the second term in Equation (1). On the other hand, the cost for our routing method becomes $O(\alpha N^2) = O(N^2)$ because it only calculates packet transmitting information along the shortest distance paths, without using v_{jl} . Thus, the proposed routing method then evaluates the optimum adjacent nodes much faster than the conventional neural-based routing method.

4. Performance Evaluation of the Proposed Method

We begin this section by evaluating our proposed routing method compared with the original MCNN routing method. Hereafter, we call the conventional MCNN routing method as the NN-O method in order to distinguish it from other neural-based routing methods. For these simulations, we adopted the same experimental assumptions and the same parameter settings as those in Section 2. We set $\zeta = 2.0$ and $\omega = 0.6$ in Equation (9).

Figure 4 illustrates the packet congestion rate for both the NN-O and the proposed method. In **Figure 4**, the proposed method is seen to maintain a lower congestion rate than the NN-O method. **Figure 5** illustrates the congestion levels of nodes for both the NN-O and the proposed method. In **Figure 5**, the proposed method can be observed to relieve packet congestion effectively at the hub nodes in comparison to the NN-O method.

Next, the performance of the proposed routing method is evaluated against three other conventional neural network based routing methods. The first is the original neural-based routing method (NN-O) [21], the second is a neural-based routing method incorporating the reinforcement learning (NN-R) [22], and the third is a neural-based routing method that includes stochastic effects (NN-S) [23].

In the neural-based routing method with reinforcement learning [22], Equation (1) is replaced by the following energy function:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^{N_i} \sum_{l=1}^{N_i} J_{ik,il} v_{ik} v_{il} - \eta_i(t) \sum_{i=1}^N \sum_{l=1}^{N_i} \left\{ 1 - \frac{1}{b_l} \left(q_l + \frac{1}{2} \sum_{j \neq i} v_{jl} \right) \right\} v_{il} - (1 - \eta_i(t)) \sum_{i=1}^N \sum_{l=1}^{N_i} \left(1 - \frac{d_l}{d_c} \right) v_{il} + \zeta \sum_{i=1}^N \left(\sum_{l=1}^{N_i} v_{il} - 1 \right)^2. \quad (10)$$

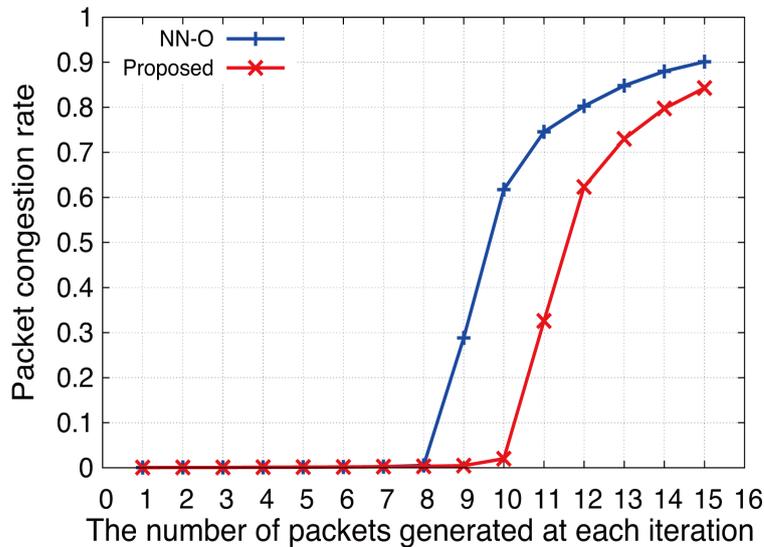


Figure 4. Packet congestion rate for NN-O and our proposed method.

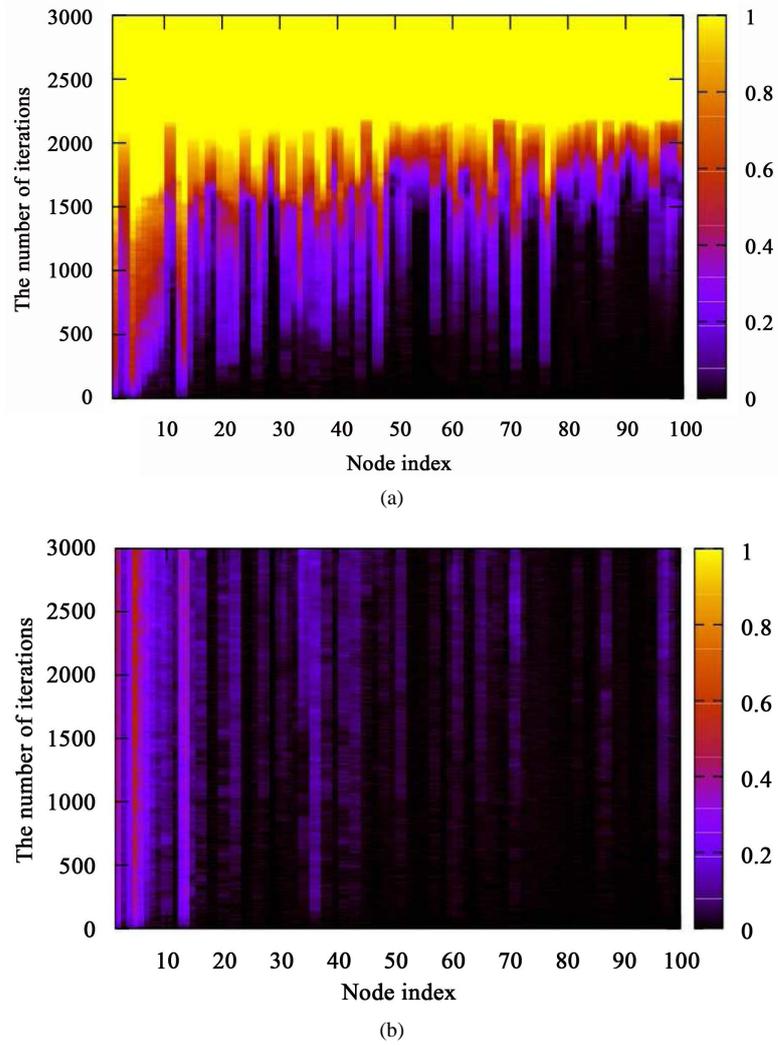


Figure 5. Congestion levels of nodes for (a) NN-O and (b) the proposed method. In this figure, we fixed the number of flowing packets to 10. The node index is sorted in descending order of the degrees of the nodes.

In Equation (10), $\eta_i(t)$ is updated according to the following algorithm:

- 1) $\eta_i(1) = C$ for any i , where C is a constant that lies between 0 and 1.
- 2) The i th node transmits a packet at the t th iteration using $\eta_i(t)$.
- 3) $\eta_i(t)$ is updated by

$$\eta_i(t) = (1 - \alpha)\eta_i(t) + \alpha \left\{ \gamma \eta_{\tilde{i}(t)}(t + r_i(t)) \right\}. \quad (11)$$

- 4) Go to 2.

In Equation (10), α is the learning rate, γ is the discount rate, and $\tilde{i}(t)$ is the index of the node to which a packet is transmitted from the i th node at the t iteration. The function $r_i(t)$ in Equation (11) is defined by

$$r_i(t) = \begin{cases} 0 & \text{if } q_{\tilde{i}(t)} < b_i(t), \\ 1 - \gamma \eta_{\tilde{i}(t)}(t) & \text{if } q_{\tilde{i}(t)} = b_i(t). \end{cases} \quad (12)$$

In this paper, we set the values of $\eta_i(1) (i = 1, \dots, N)$ in Equation (10) to 0.5, and we set $\alpha = 0.1$ and $\gamma = 0.9$ in Equation (12). These values are identical to those in reference [22].

In the neural-based routing method including stochastic effects [23], $\eta_i(t)$ in Equation (10) is replaced by

$$\eta_i(t) = \text{rand}[0.0, 1.0]. \quad (13)$$

In these simulations, β and ξ are set to 3.0 and 1.0 respectively for all the methods, and we set $\zeta = 2.0$ and $\omega = 0.6$ in Equation (9).

Figure 6 illustrates the packet congestion rate for each of the routing methods. In Figure 6, the packet congestion rate for the NN-O, NN-R, and NN-S methods increases more quickly than for the proposed method. The congestion rate for the proposed method increases once the number of packets generated at each iteration increases to 10. Thus, the proposed method is able to inhibit the occurrence of the packet congestion as the number of packets flowing in the network increases.

Next, we evaluate the average numbers of arriving and lost packets, using a computer network model in which a fixed number of the packets flow [23] [24] [28]. The average number of arriving packets, \bar{A} , and the average number of lost packets, \bar{L} , are defined by

$$\bar{A} = \frac{N_a}{N_g}, \bar{L} = \frac{N_l}{N_g}, \quad (14)$$

where N_a is the number of arriving packets, N_g is the number of generated packets, and N_l is the number of lost packets.

Figure 7 illustrates the average numbers of arriving and lost packets for NN-O, NN-R, NN-S, and the proposed method. In Figure 7(a), the proposed method is seen to achieve a higher arrival rate for packets if the average number of packets at each node is between 10 and 40. In addition, the proposed method achieves the lowest lost packet rate of all the neural-based routing methods if the average number of packets at each node is less than 40. Thus, we have clarified that the proposed routing method achieves a better performance than the other neural-based routing methods if the number of packets flowing in the computer networks is small. The above results suggest that the transmitting information provides an effective method of avoiding packet congestion in order to reliably route packets. If the average number of packets at each node is larger than 50, then almost all of the nodes in the network will be holding more than 50% of the number of packets needed to fill their buffer. Thus, in this case packet congestion appears throughout the computer network, and any routing method will perform less effectively in removing congestion.

5. Conclusions

In this paper, we use the packet congestion rate to evaluate a proposed packet routing method that employs mutually connected neural network and transmitting information. The packet congestion rate is mainly employ-

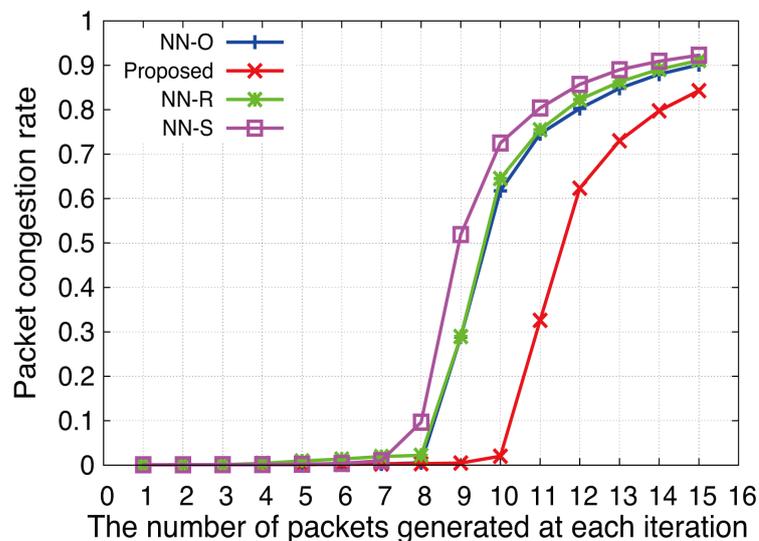
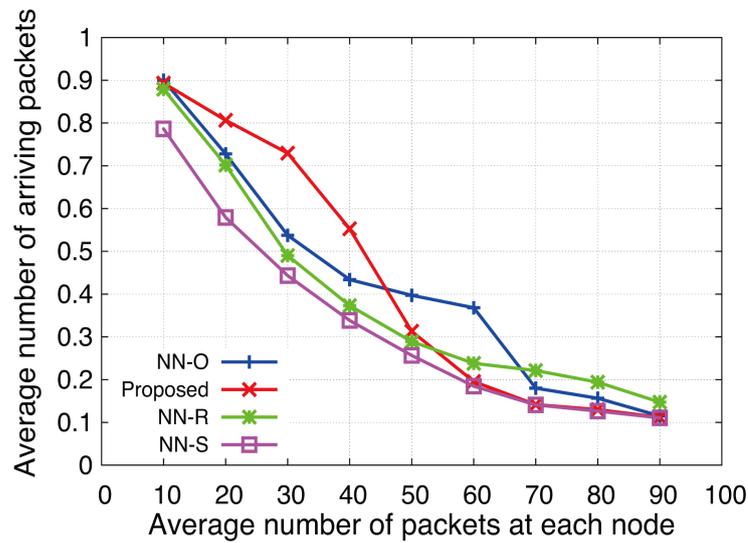
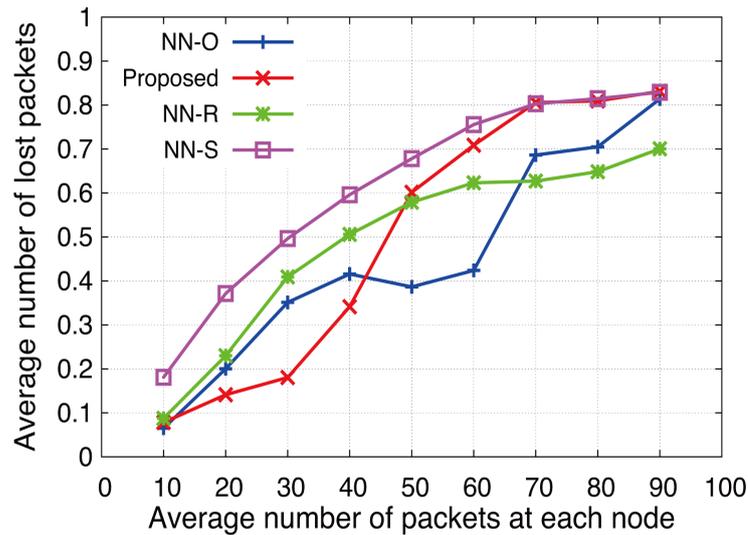


Figure 6. Packet congestion rates of the routing methods for the scale-free networks.



(a)



(b)

Figure 7. Relationship between the average number of packets at each node and (a) the average number of arriving packets and (b) the average number of lost packets.

ed to clarify performance in terms of congestion avoidance as the number of flowing packets gradually increases in a communication network. First, we show that the original neural-based routing method is unable to alleviate packet congestion, because packets accumulate at hub nodes of large degree in the network, and packet congestion is then distributed into the network. To overcome this problem, we propose a neural-based routing method that exploits transmitting information, and evaluate our proposed method against conventional neural-based routing methods, which incorporate either reinforcement learning or stochastic effects. From the results of the numerical simulations, we can conclude that our proposed method is able to reduce the packet congestion rate more effectively than the other neural-based routing methods. We also evaluate our proposed method in terms of the average numbers of arriving and lost packets by using a computer network model in which a fixed number of packets flow. We are able to confirm that our proposed method can maintain a higher arrival rate, or lower lost rate, than the other neural-based routing methods. From these results, we conclude that our proposed neural-based routing method achieves reliable communication by avoiding the congestion of packets through the effective use of transmitting information.

For this study, we assumed that information regarding the number of packets accumulated on a path could instantaneously be obtained by all nodes. In future works, we will consider how to deal with delays in this information in order to implement the proposed neural-based routing method into real-world systems.

The research of T.K. was partially supported by a Grant-in-Aid for Young Scientists (B) from JSPS (No. 16K21327).

References

- [1] Kai, C.H., Long, J., Ling, X. and Hu, M.B. (2014) Upper Bound of Network Capacity and a Static Optimal Packet Routing Strategy. *Physica A*, **401**, 174-181. <http://dx.doi.org/10.1016/j.physa.2014.01.045>
- [2] Tan, F. and Xia, Y. (2013) Hybrid Routing on Scale-Free Networks. *Physica A*, **392**, 4146-4153. <http://dx.doi.org/10.1016/j.physa.2013.04.032>
- [3] Ling, X., Hu, M.-B., Jiang, R. and Wu, Q.-S. (2010) Global Dynamic Routing for Scale-Free Networks. *Physical Review E*, **81**, 016113. <http://dx.doi.org/10.1103/PhysRevE.81.016113>
- [4] Ling, X., Hu, M.-B., Jiang, R., Wang, R., Cao, X.-B. and Wu, Q.-S. (2009) Pheromone Routing Protocol on a Scale-Free Network. *Physical Review E*, **80**, 066110. <http://dx.doi.org/10.1103/PhysRevE.80.066110>
- [5] Wang, D., Jing, Y. and Zhang, S. (2008) Traffic Dynamics Based on a Traffic Awareness Routing Strategy on Scale-Free Networks. *Physica A*, **387**, 3001-3007. <http://dx.doi.org/10.1016/j.physa.2008.01.085>
- [6] Wu, J.-J., Sun, H.-J. and Gao, Z.-Y. (2008) Dynamic Urban Traffic Flow Behavior on Scale-Free Networks. *Physica A*, **387**, 653-660. <http://dx.doi.org/10.1016/j.physa.2007.09.020>
- [7] Zhang, H., Liu, Z., Tnag, M. and Hui, P. (2007) An Adaptive Routing Strategy for Packet Delivery in Complex Networks. *Physics Letters A*, **364**, 177-182. <http://dx.doi.org/10.1016/j.physleta.2006.12.009>
- [8] Hu, M.-B., Wang, W.-X., Jiang, R., Wu, Q.-S. and Wu, Y.-H. (2007) Phase Transition and Hysteresis in Scale-Free Network Traffic. *Physical Review E*, **75**, 036102. <http://dx.doi.org/10.1103/PhysRevE.75.036102>
- [9] Yan, G., Zhou, T., Hu, B., Fu, Z.-Q. and Wang, B.-H. (2006) Efficient Routing on Complex Networks. *Physical Review E*, **73**, 046108. <http://dx.doi.org/10.1103/PhysRevE.73.046108>
- [10] Wang, W.-X., Wang, B.-H., Yin, C.-Y., Xie, Y.-B. and Zhou, T. (2006) Traffic Dynamics Based on Local Routing Protocol on a Scale-Free Network. *Physical Review E*, **73**, 026111. <http://dx.doi.org/10.1103/PhysRevE.73.026111>
- [11] Wang, W.-X., Yin, C.-Y., Yan, G. and Wang, B.-H. (2006) Integrating Local Static and Dynamic Information for Routing Traffic. *Physical Review E*, **74**, 016101. <http://dx.doi.org/10.1103/PhysRevE.74.016101>
- [12] Wang, W.-X., Wang, B.-H., Hu, B., Yan, G. and Ou, Q. (2005) General Dynamics of Topology and Traffic on Weighted Technological Networks. *Physical Review Letters*, **94**, 188702. <http://dx.doi.org/10.1103/PhysRevLett.94.188702>
- [13] Zhao, L., Lai, Y.-C., Park, K. and Ye, N. (2005) Onset of Traffic Congestion in Complex Networks. *Physical Review E*, **71**, 026125. <http://dx.doi.org/10.1103/PhysRevE.71.026125>
- [14] Echenique, P., Gómez-Gardenes, J. and Moreno, Y. (2005) Dynamics of Jamming Transitions in Complex Networks. *Europhysics Letters*, **71**, 325-331. <http://dx.doi.org/10.1209/epl/i2005-10080-8>
- [15] Echenique, P., Gómez-Gardenes, J. and Moreno, Y. (2004) Improved Routing Strategies for Internet Traffic Delivery. *Physical Review E*, **70**, 056105. <http://dx.doi.org/10.1103/PhysRevE.70.056105>
- [16] Arenas, A., Díaz-Guilera, A. and Guimerà, R. (2001) Communication in Networks with Hierarchical Branching. *Physical Review Letters*, **86**, 3196-3199. <http://dx.doi.org/10.1103/PhysRevLett.86.3196>
- [17] Ohira, T. and Sawatari, R. (1998) Phase Transition in a Computer Network Traffic Model. *Physical Review E*, **58**, 193-195. <http://dx.doi.org/10.1103/PhysRevE.58.193>
- [18] Dijkstra, E.W. (1959) A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, **1**, 269-271. <http://dx.doi.org/10.1007/BF01386390>
- [19] Bellman, R.E. (1958) On a Routing Problem. *Quarterly of Applied Mathematics*, **16**, 87-90.
- [20] Floyd, R.W. (1962) Algorithms 97: Shortest Path. *Communications of the ACM*, **5**, 345. <http://dx.doi.org/10.1145/367766.368168>
- [21] Horiguchi, T. and Ishioka, S. (2001) Routing Control of Packet Flow Using a Neural Network. *Physica A: Statistical Mechanics and Its Applications*, **297**, 521-531. [http://dx.doi.org/10.1016/S0378-4371\(01\)00229-1](http://dx.doi.org/10.1016/S0378-4371(01)00229-1)
- [22] Horiguchi, T., Hayashi, K. and Tretiakov, A. (2005) Reinforcement Learning for Congestion-Avoidance in Packet Flow. *Physica A: Statistical Mechanics and Its Applications*, **349**, 329-348. <http://dx.doi.org/10.1016/j.physa.2004.10.015>

- [23] Kimura, T., Nakajima, H. and Ikeguchi, T. (2007) A Packet Routing Method for Complex Networks by a Stochastic Neural Network. *Physica A: Statistical Mechanics and Its Applications*, **376**, 658-672. <http://dx.doi.org/10.1016/j.physa.2006.10.061>
- [24] Kimura, T., Takamizawa, T. and Jin'no, K. (2015) Neural-Based Routing Strategy with Transmission Information for Complex Communication Networks. *Nonlinear Theory and Its Applications*, **6**, 263-274. <http://dx.doi.org/10.1587/nolta.6.263>
- [25] Hopfield, J.J. and Tank, D.W. (1985) Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics*, **52**, 141-152.
- [26] Faloutsos, M., Faloutsos, F. and Faloutsos, C. (1999) On Power-Law Relationships of the Internet Topology. *Computer Communication Review*, **29**, 251-262. <http://dx.doi.org/10.1145/316194.316229>
- [27] Barabási, A.L. and Albert, R. (1999) Emergence of Scaling in Random Networks. *Science*, **286**, 509-512. <http://dx.doi.org/10.1126/science.286.5439.509>
- [28] Kimura, T., Ikeguchi, T. and Tse, C.K. (2012) Efficient Routing Strategy with Memory Information for Complex Networks. *American Journal of Operations Research*, **2**, 73-81. <http://dx.doi.org/10.4236/ajor.2012.21008>



Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>