RESEARCH ARTICLE

A routing method with adaptively adjusting memory information based on local routing history

Takayuki Kimura[1]*, **Yutaka Shimada**[2]

1 Department of Electrical, Electronics and Communication Engineering, Faculty of Fundamental Engineering, Nippon Institute of Technology, Miyashiro, Saitama, Japan, **2** Department of Information and Computer Sciences, Graduate School of Sciences and Engineering, Saitama University, Saitama, Japan

* tkimura@nit.ac.jp

OPEN ACCESS

Citation: Kimura T, Shimada Y (2023) A routing method with adaptively adjusting memory information based on local routing history. PLoS ONE 18(4): e0283472. https://doi.org/10.1371/journal.pone.0283472

Editor: Chakchai So-In, Khon Kaen University, THAILAND

Received: September 2, 2022

Accepted: March 8, 2023

Published: April 19, 2023

Data Availability Statement: Data are available on GitHub (https://github.com/tkimura1113/dataset_for_memory_routing.git).

Funding: - Initials of the authors who received each award TK YS - Grant numbers awarded to each author TK: 19K04395 JP21H03514 YS: JP21H03514 JP21H03508 - The full name of each funder Japan Society for the Promotion of Science - URL of each funder website https://www.jsps.go.jp/english/index.html The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Finding the shortest paths for packets from sources to destinations in packet-switched communication networks is an inevitable problem in building a future high-speed information society. A routing method with memory information has already been proposed to alleviate the congestion of large volumes of packet flow. This routing method shows a high transmission completion rate even for large volumes of packet flows in communication networks with scale-free properties. However, the method exhibits poor performance for networks with local triangular connections and long distances between nodes. To overcome these problems, in this study, we first enhanced the routing performance of the conventional communication network models by using the betweenness centrality of nodes, which is one of the network centralities that measures the number of shortest paths that pass through each node in the networks. Subsequently, we adaptively changed the transmitting paths of packets by using only local information. Numerical simulations indicated that our routing method performs successfully for various topologies of communication networks by avoiding the congested node, and effectively using the memory information.

Introduction

The recent significant development of communication technologies, such as the Internet of Things, various applications using cloud communication, and the fifth generation communication will result in a huge amount of traffic to the packet-switched communication networks. The shortest path routing (SP) method is commonly employed as a routing method in real communication networks. This method successfully handles the transmission of small volumes of traffic flows in the communication networks from their sources to destinations. However, for large traffic volumes, the transmission of data to their destinations is hindered because of existing scale-free (SF) properties in real communication networks [1]; the packets are easily congested at routers, where many transmitting paths go through. One of the solutions to avoid traffic congestion is to rewire the physical connections between any hosts or routers. However, this requires huge costs. The second solution is to implement a sophisticated

Competing interests: The authors have declared that no competing interests exist.

routing method that effectively diversifies the transmitting routes for packets depending on the traffic volumes and the underlying topologies in the communication networks.

Considering the conventional routing methods that aim to address data transmission in high-traffic communication networks, Echenique *et al.* [2] proposed a routing method that employs distance information and the number of stored packets at nodes. Ling *et al.* [3] proposed a global dynamic routing that uses stored information of packets in the entire communication network. Tang *et al.* [4] proposed a self-adjusting traffic awareness protocol that utilizes the connectivity of communication networks and the stored information of packets. Wang *et al.* [5, 6] analyzed the traffic dynamics of communication networks with SF properties and proposed a routing algorithm that integrates the local static and dynamic information. Yang *et al.* [7, 8] proposed an effective routing method for mobile networks. Gao *et al.* [9] employed weights between nodes that dynamically change based on the connectivity of the network and stored information of packets. Echagüe *et al.* [10, 11] proposed a routing algorithm that dynamically changes the distance between nodes using congestion parameters based on the number of packets that arrive at the nodes. Lin *et al.* [12] proposed a routing strategy with a restrictive queue length algorithm, which reduces the load of the hub nodes in the networks. Zhang *et al.* [13] proposed the routing strategy in which each packet at a node has a different level of priority for the next transmitting node. Horiguchi *et al.* [14] proposed a routing strategy by using the Hopfield and Tank typed mutually connected neural networks, and this method was further improved by reinforcement of a learning strategy [15] and incorporation of the stochastic effect [16]. Recently, Hou *et al.* proposed QoS routing algorithms based on the particle swarm optimization [17] and the ant colony optimization [18] for different types of services in named data networking, which are the next generation of network architectures. The above-mentioned routing methods [2, 9–11, 14–16] require real-time information such as the number of stored packets at adjacent (or connected) nodes to determine the next transmitting node. However, such information cannot always and instantaneously be obtained in real networks. Therefore, a routing method that avoids packet congestion using only local information is required.

From this viewpoint, a routing method that autonomously diversifies packet transmissions using chaotic neurodynamics has been proposed [19, 20]. An advantage of using chaotic neurodynamics is the refractory effect, which inhibits the firing of neurons for a certain period of time [21]. By using this functionality, the solution methods with the chaotic neurodynamics were applied to various types of combinatorial optimization problems [22]. In the routing method using the chaotic neurodynamics, the refractory effect works to memorize the histories of packet transmissions at each node. Further, a routing method using memory information, which reinforces the packet memorizing functionality in chaotic neural networks, has already been proposed [23]. According to the hop distance information and memory information, this routing method [23] avoids the congestion of packets and maintains a high transmission completion rate of packets for the communication networks with SF properties. In addition, the effectiveness of the memory information is clarified by using the method of surrogate data [24], which is a statistical hypothesis testing used in the research field of nonlinear time-series analysis [25]. However, the routing method using memory information showed poor performance for networks with highly clustered and long distances between nodes [24].

In this work, we incorporated two strategies to improve the routing method using memory information [23, 24]. First, we reformed the conventional communication network models [26] by changing the number of transmissions of packets and buffer sizes at each node based on node betweenness centrality, which is one of the network centralities that measures the number of shortest paths that pass through nodes in the networks [27]. Second, we enhanced the performance of the routing method [24] by adaptively adjusting the weight of memory information at each node based on the transmitting probabilities. Numerical experiments indicated that our proposed routing method that adaptively adjusted the weights of the memory information worked well both for communication networks with SF properties and also highly clustered networks with long node distances.

Communication network models

This study used communication network models represented by an unweighted and undirected graph $U = (V, E)$, where V is a set of nodes, and E is a set of edges [2, 4, 9–11, 20, 24, 28, 29]. In the communication network model, each node represents a host and a router, and each edge represents a connection between the nodes. A packet is generated at a randomly selected node (source), and a randomly selected destination different from the source is assigned. Each node has a buffer for storing packets. If a packet is generated at a node, it is stored at the tail of the buffer of the node. A packet is removed from the network if it is transmitted to a node with a full volume of the packets in its buffer or in case the packet arrives at its destination. All packets are transmitted according to the first-in-first-out principle.

We first enhanced the routing performance of communication network models proposed in Ref. [26] by improving the number of transmissions of packets and buffer sizes at each node. To show that the conventional communication network models [23, 24, 26] have poor packet routing abilities, we first displayed relationships between degree and node betweenness centralities of networks and the number of existing packets by using the *shortest path* (SP) method. The SP method transmits packets from sources to destinations using uniquely determined shortest paths. In this work, if multiple adjacent nodes have the same shortest hop distance to the designations, we assigned an adjacent node that has the youngest index as the transmitting node of packets. The shortest paths for all pairs of nodes are preliminarily obtained once the networks were constructed using the shortest path algorithm, such as the Dijkstra algorithm [30].

In Ref [26], the storage performance of node i that corresponds to the buffer sizes at each node, B_i, is defined by:

$$B_i = \beta k_i, \tag{1}$$

where $\beta > 0$ is a control parameter and k_i is the degree of node i, which corresponds to the number of edges of a node. By using Eq (1), each node has space for storing packets, which is proportional to its degree. Next, the transmission performance of node i that corresponds to the number of transmissions of packet at each node, C_i, is defined as:

$$C_i = 1 + \lfloor \gamma k_i + 0.5 \rfloor, \tag{2}$$

where $\gamma > 0$ is a control parameter and $\lfloor \cdot \rfloor$ expresses the floor function. By using Eq (2), the node that has a large degree transmits many packets to its adjacent (connected) nodes during one iteration. In this work, one iteration is defined as a period for each node to transmit the first C_i, $\forall i \in V$ packets to the adjacent nodes.

Next, we evaluated the SP method for the conventional and improved communication network models. In all the numerical experiments, we set the number of iterations, denoted by I, as $I = 10^3$. In addition, we set β and γ to 10^3 and 0.4, respectively, based on conventional studies [23, 24]. We also generated R packets at each iteration.

We generated SF networks, small-world networks, and networks with both scale-free and small-world properties by using the Barabási and Albert (BA) [1], the Watts and Strogatz (WS) [31], and the Klemm and Eguíluz (KE) [32] models, respectively. The BA model [1] was constructed using the following procedure. We started with a complete graph of m nodes, and inserted a new node with m edges at every time step. Next, we connected the m edges of the newly added node to the nodes that already existed in the network with probability $\Pi(k_i) = k_i^{(t)} / \sum_{m=1}^{|V^{(t)}|} k_m^{(t)}$, where $k_i^{(t)}$ is the degree of node i ($i = 1, \ldots, |V^{(t)}|$) at time t, $V^{(t)}$ is a set of indices of pre-existing nodes in a network at time t, and $|\cdot|$ denotes the number of elements in a set.

For the WS model [31], we started from a circular regular network with eight degrees in each node. Next, we randomly rewired each edge with a probability r_p ($0 \leq r_p \leq 1$) so that the networks were strongly connected. This construction allowed us to adjust the network between the regular network ($r_p = 0$), small-world network ($0 < r_p < 1$), and the fully randomized network ($r_p = 1$).

The KE model [32] was constructed following a procedure similar to that for the BA model. In this model, each node had either an activated or a deactivated state, and we set a newly added node with m edges to an activated state. We started with a complete graph of m nodes with activated states. The m edges of the newly added node were connected to the following criteria; (i) m edges connect to the activated nodes, (ii) m edges connect to the existing nodes, including both activated and deactivated nodes using the preferential attachment whose probability was the same as that used to construct the BA model. The criterion (i) is selected with the probability $\mu(0 \leq \mu \leq 1)$, and (ii) is selected with probability $1 - \mu$ [32]. Next, one of the activated nodes was deactivated. The probability of choosing node i for deactivation is inversely proportional to its degree. In this study, the number of edges at each node for the regular networks ($r_p = 0$) of the WS model was set to 8, and m of the BA and KE models was set to 4.

Next, we normalized the node betweenness centrality, degree centrality, and the number of existing packets. To calculate the normalized node betweenness and normalized degree centralities, we first measured the node betweenness centrality of node i, χ_i, and degree centrality of node i, f_i. Those are defined as follows:

$$\chi_i = \frac{\displaystyle\sum_{i_s=1; i_s \neq i}^{|V|} \sum_{i_t=1; i_t \neq i}^{i_s-1} \frac{g_i^{(i_s,i_t)}}{N_{i_s,i_t}}}{(|V|-1)(|V-2|)/2}, \quad i \in V, \tag{3}$$

$$f_i = \frac{k_i}{\sum_{j=1}^{|V|} k_j}, \quad i \in V, \tag{4}$$

where $g_i^{(i_s,i_t)}$ is the number of shortest paths from node i_s to i_t that pass through node i, N_{i_s,i_t} is the total number of shortest paths from node i_s to i_t, and k_i is the degree of node i. The node betweenness centrality measures how often a node is on the shortest paths between any pairs of nodes in the networks, and the degree centrality measures the number of edges attached to the nodes. Next, we prepared vectors whose elements were the node betweenness and the degree centralities of nodes defined by $\boldsymbol{X} = (\chi_1, \ldots, \chi_{|V|})^{\top}$ and $\boldsymbol{F} = (f_1, \ldots, f_{|V|})^{\top}$, respectively. We then created a normalized vector by $\boldsymbol{X}/||\boldsymbol{X}||$ and $\boldsymbol{F}/||\boldsymbol{F}||$, where $|| \cdot ||$ denotes the Euclidean norm of a vector. To obtain normalized the number of existing packets, we counted the number of stored packets of node i during the simulation as $Q_i = \sum_{t=1}^{I} q_i(t)$, where $q_i(t)$ is the number of stored packets of the node i at the iteration t, and calculated the number of existing packet vectors as $\boldsymbol{Q} = (Q_1, \ldots, Q_{|V|})^{\top}$. $\boldsymbol{Q}$ was then normalized by $\boldsymbol{Q}/||\boldsymbol{Q}||$.

Fig 1 shows scatter plots between the normalized degree centrality and normalized the number of existing packets (upper figures) and scatter plots between the normalized node betweenness centrality and normalized the number of existing packets (lower figures) for the BA, WS, and KE models. Not surprisingly, in Fig 1(d)–1(f), the normalized node betweenness

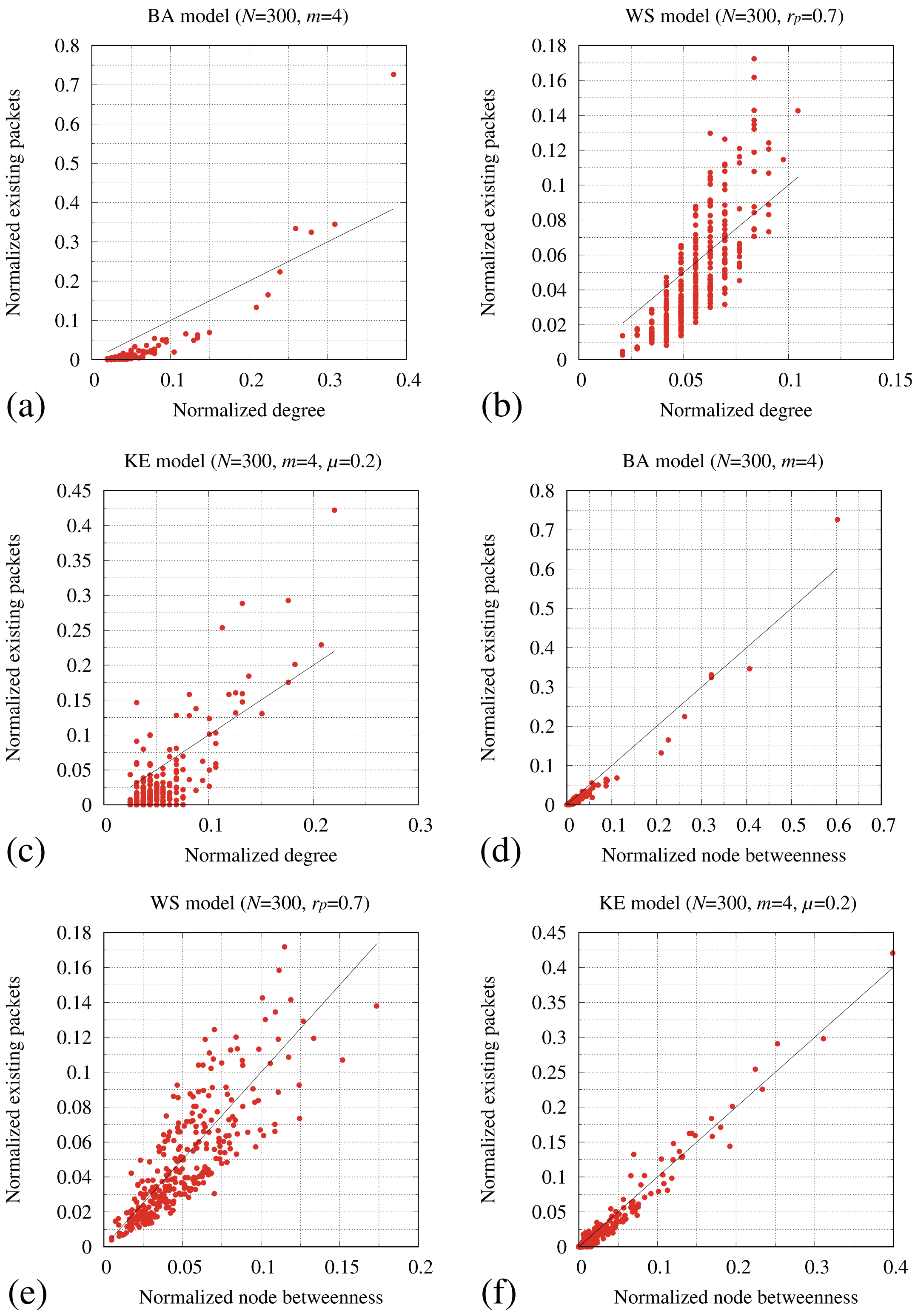

Fig 1. **Upper figures**: scatter plots of the normalized degree centrality of networks and normalized the number of existing packets generated by the SP method for the (a) BA, (b) WS, and (c) KE models. **Lower figures**: scatter plots of the normalized betweenness centrality of networks and normalized the number of existing packets generated by the SP method for the (d) BA, (e) WS, and (f) KE models. Black lines in all figures represent $f(x) = x$.

https://doi.org/10.1371/journal.pone.0283472.g001

centrality and normalized the number of existing packets exhibit a stronger correlation than that of the normalized degree centrality (Fig 1(a)–1(c)), particularly in the KE model (Fig 1(c) and 1(f)). These results strongly suggest that B_i and C_i are defined based on the node betweenness centrality [27]. Thus, we modified the conventional packet transmission and storage performance of node i, expressed by $\tilde{B}_i, \tilde{C}_i, \forall i \in V$ as follows:

$$\tilde{B}_i = \tilde{B}_{\min} + B_{\text{all}}\tilde{b}_i, \tag{5}$$

$$\tilde{C}_i = \tilde{C}_{\min} + \lfloor C_{\text{all}}\tilde{b}_i + 0.5 \rfloor, \tag{6}$$

where $\tilde{b}_i$ is the normalized node betweenness centrality of node i defined by $\tilde{b}_i = b_i / \sum_{j=1}^{|V|} b_j$, and b_i is the node betweenness centrality of node i. B_{all} and C_{all} are control parameters that adjust $\sum_{i=1}^{|V|} B_i \approx \sum_{i=1}^{|V|} \tilde{B}_i$ and $\sum_{i=1}^{|V|} C_i \approx \sum_{i=1}^{|V|} \tilde{C}_i$ and are defined by $B_{\text{all}} = \sum_{i=1}^{|V|} (B_i - B_{\min}) \geq 0$ and $C_{\text{all}} = \sum_{i=1}^{|V|} (C_i - C_{\min}) \geq 0$, respectively. $\tilde{B}_{\min}$ and $\tilde{C}_{\min}$ guarantee that the node i has $\tilde{B}_{\min}$ and $\tilde{C}_{\min}$ at the minimum. We set $B_{\min}$ and $C_{\min}$ to 7 for all numerical simulations.

In sequence, we evaluated the transmission completion rate of packets for the conventional and improved communication network models. The transmission completion rate of packets is described as follows:

1. Transmission completion rate of packets A:

$$A = \frac{1}{RI} \sum_{t=1}^{I} a(t), \tag{7}$$

where $a(t)$ is the number of packets that arrive at the destination at the tth iteration. If A is 1, all the packets are successfully transmitted to their destinations.

Fig 2 shows the transmission completion rate of packets (A) defined by Eq (7) by the SP method for the conventional and improved transmission and storage performance of nodes. In these experiments, we constructed 30 different networks for every BA, WS, and KE model and averaged the results. These results indicated that the SP method maintained a higher A for the improved communication network in all the topologies of networks. From these results, we hereafter used the packet transmission and storage performance of nodes defined by Eqs (5) and (6) for later numerical experiments.

Routing method using memory information

As our proposed routing method improves the conventional routing method using memory information [23, 24], we first describe this routing method and evaluate it for improved communication network models. To implement the routing method using memory information (hereafter referred to as the *memory* method), we first define the distance information, ξ_{ij}, as follows:

$$\xi_{ij} = \frac{d_{ij} + d_{jg(\pi_i)}}{\sum_{s \in \partial_i} (d_{is} + d_{sg(\pi_i)})} = \frac{1 + d_{jg(\pi_i)}}{\sum_{s \in \partial_i} (1 + d_{sg(\pi_i)})}, \qquad i \in V,\ j \in \partial_i, \tag{8}$$

where d_{ij} is the static shortest hop distance between the node i and its adjacent node j, π_i is a packet ready to be transmitted from node i, $g(\pi_i)$ is the destination node of π_i, $d_{jg(\pi i)}$ is the static shortest hop distance from the adjacent node j to $g(\pi_i)$, and ∂_i is the set of adjacent nodes of

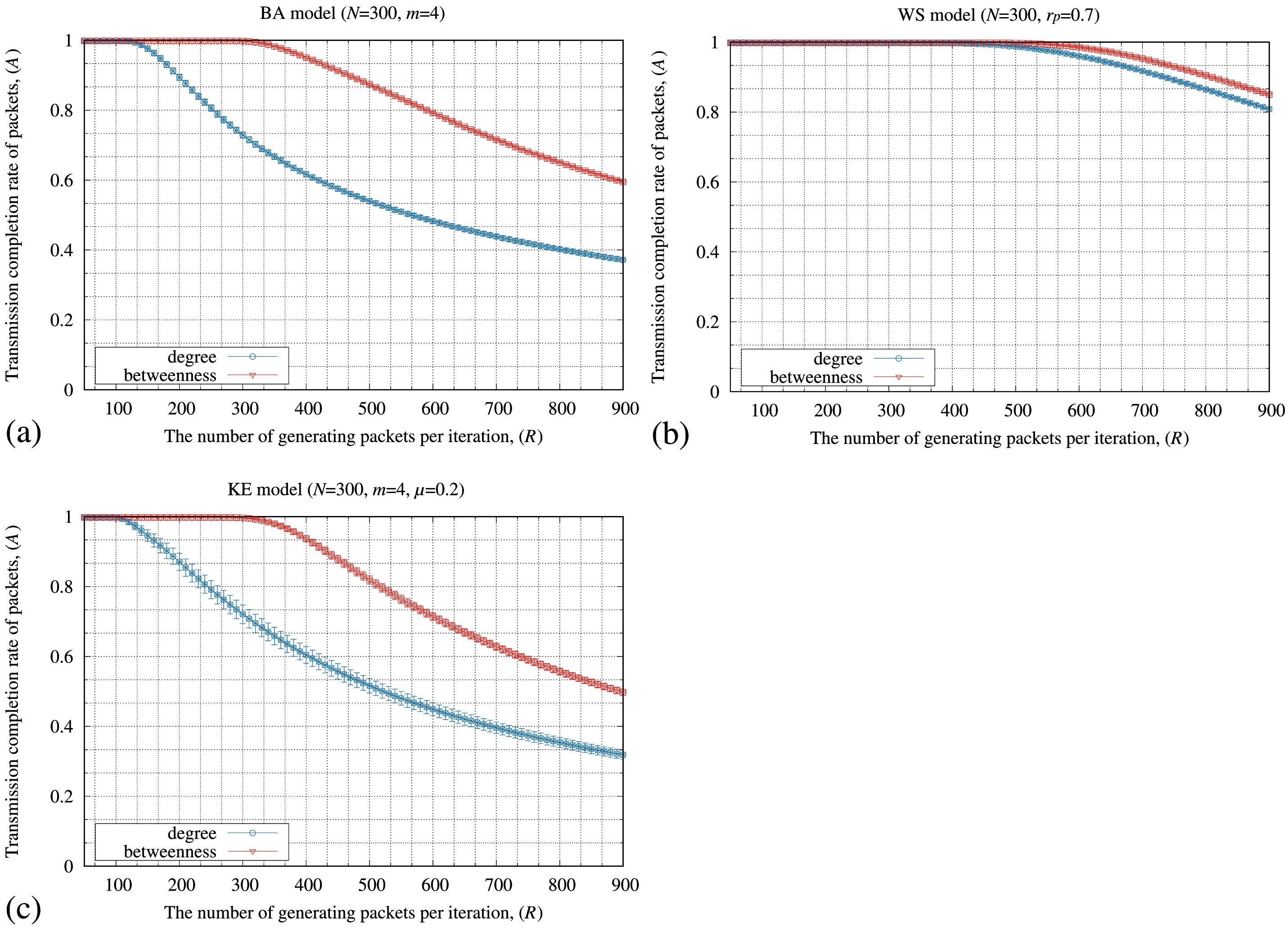

Fig 2. Relationship between the number of generating packets at each iteration (R) of the SP method and the transmission completion rate of packets (A) defined by Eq (7) for the (a) BA, (b) WS, and (c) KE models. In these figures, the "degree" and "betweenness" in the figure legends express that the packet transmission and storage performance are composed of the node degree and node betweenness centralities, respectively. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g002

node i. The SP method is realized by using the distance information only; node i transmits a packet to its adjacent node j with the minimum value of ξ_{ij}, namely $\min_{j\in\partial_i}\xi_{ij}$.

Next, the memory information, $\zeta_{ij,\tau}$, is defined as follows:

$$\zeta_{ij,\tau} = \alpha\sum_{s=0}^{\tau}\gamma^{s}x_{ij}(\tau - s) = \alpha x_{ij}(\tau) + \gamma\zeta_{ij,\tau-1}, \qquad i \in V,\ j \in \partial_i,\ 1 \le \tau \le \omega_{ij} \tag{9}$$

where $\alpha > 0$ is a control parameter determining the strength of the memory information, $\gamma(0 < \gamma \le 1)$ is a decay parameter of the memory information, ω_{ij} is the total number of transmissions of packets from node i to j, and $x_{ij}(\tau)$ is a memorizing variable of transmission from node i to j at the τth packet transmission defined as follows:

$$x_{ij}(\tau) = \begin{cases} 1 & (\text{if a node } i \text{ transmits a packet to the node } j \text{ at the } \tau\text{th packet transmission}), \\ 0 & (\text{otherwise}). \end{cases} \tag{10}$$

Using the distance and memory information, we defined an evaluating function to determine the transmission of a packet from node i to its adjacent node j as follows:

$$\min_{j \in \partial_i} y_{ij,\tau}, \qquad i \in V, \ 1 \leq \tau \leq \omega_{ij}, \tag{11}$$

where $y_{ij,\tau} = \xi_{ij} + \zeta_{ij,\tau}$, and $\partial_i \subseteq V$ is a set of adjacent nodes of node i.

In this method, the value of $\zeta_{ij,\tau}$ increases if node i frequently transmits the packets to the adjacent node j. Therefore, node i avoids to transmit the next packet to the adjacent node j for a while because the smallest value of $y_{ij,\tau}$ in Eq (11) is determined as the next transmitting node. Thus, the memory information memorizes the past transmitting history at each node. Moreover, this method does not require additional information on the adjacent nodes, such as the number of stored packets, to diversify transmitting routes for packets.

We compared the transmission completion rate of packets by the memory, SP, *random SP* (SPr) methods, and the routing method by using the distance and packet stored distribution proposed in [2] for the networks generated from the BA, WS, and KE models. The SP method transmits a packet using uniquely determined shortest paths. In this work, we fixed indices of nodes after constructing a network. A node in the SP method transmits a packet to its adjacent node with the youngest index when multiple adjacent nodes satisfy $\min_{j \in \partial_i} \xi_{ij}$. Moreover, a node in the SPr method randomly transmits a packet to one of its adjacent nodes with the smallest value of ξ_{ij}. The routing method [2] determines the transmitting nodes of packets using the following equation:

$$\Omega_{ij}(t) = \psi(d_{ij} + d_{jg(\pi_i)}) + (1 - \psi)q_j(t), \ i \in V, j \in \partial_i, \tag{12}$$

where $q_j(t)$ is the number of stored packets in the adjacent node j at the tth iteration, and $\psi(0 < \psi < 1)$ is a tunable parameter that determines the strength of the first and second terms. In the right-hand side of Eq (12), the first term is the shortest hop distance for the packet from node i to the destination of the packet through the adjacent node j, and the second term is the number of stored packets in the adjacent node j. In this method, the node i transmits a packet to its adjacent node j that has the smallest value of $\Omega_{ij}(t)$. Our preliminary numerical experiments show that the performance of the routing method [2] is rapidly degraded if R becomes large because the values of ranges between the first and second terms are too different (see S1 Fig in S1 File for details); Eq (12) is dominated by the second term because $d_{ij} + d_{jg(\pi_i)} \ll q_j(t)$ if R increases enough. To overcome this problem, we modified Eq (12) as follows:

$$\Omega'_{ij}(t) = \psi\left[\frac{d_{ij} + d_{jg(\pi_i)}}{\sum_{s \in \partial_i}(d_{is} + d_{sg(\pi_i)})}\right] + (1 - \psi)\frac{q_j(t)}{\sum_{s \in \partial_i} q_s(t)}, \ i \in V, j \in \partial_i. \tag{13}$$

We preliminarily conducted the numerical simulations to determine ψ for this routing method, and ψ was set to 0.9. Hereafter, we call the routing method using Eq (13) the *efficient routing*(*ER*) method.

In these numerical experiments, we set the number of iterations to $I = 10^3$. In the memory method, we set α and γ in Eq (9) to 0.01 and 0.99 [24], respectively. We constructed 30 different networks for every BA, WS, and KE model and averaged the results.

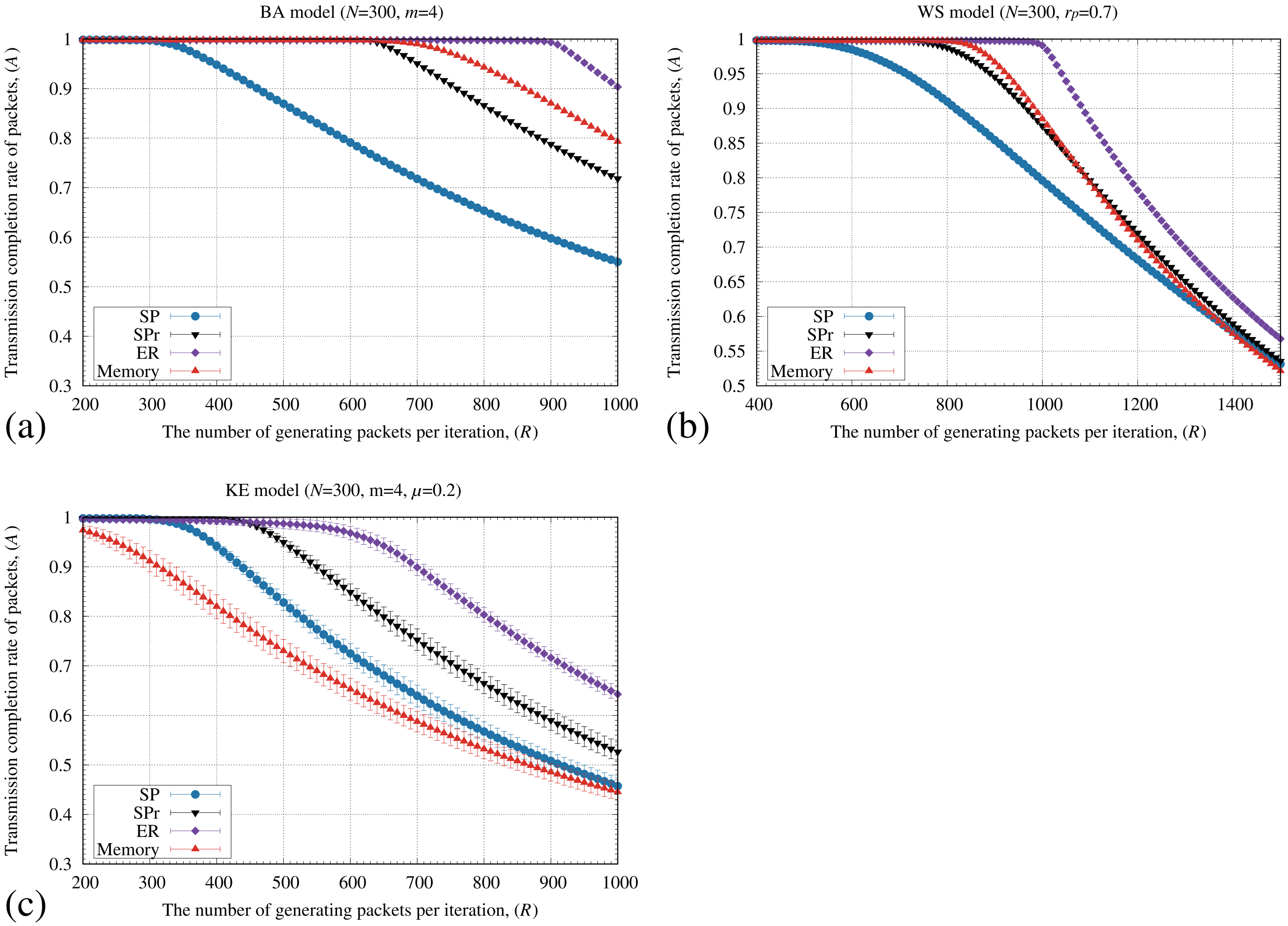

Fig 3. Relationship between the number of generating packets at each iteration (R) and transmission completion rate of packets (A) by the SP, SPr, ER, and memory methods for the (a) BA, (b) WS, and (c) KE models. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g003

Fig 3 illustrates the transmission completion rate of packets (A) by the SP, SPr, ER, and memory methods for the (a) BA, (b) WS, and (c) KE models as the number of generating packets (R) increased. In Fig 3(a) and 3(b), all the routing methods keep 100% of A if R is a small value. In addition, the value of R at which A decreased from 100% by the memory method is more significant than those of the SP and SPr methods for the BA and WS models. However, the memory method shows the lowest A in the KE model (Fig 3(c)). Although the memory method used the information of the transmitting node such as the distance and memory information, the ER method used information of both the transmitting and adjacent nodes such as the distance and the number of stored packets at adjacent nodes for each transmission of a packet. The ER method then retains the highest A value for all the network models. One of the reasons why the memory method shows poor performance for the KE model is triangular connections in the networks. For example, if nodes a, b, and c had a triangular connection, these nodes are mutually connected to each other. In this case, a packet starting from a is transmitted to c by using the following two paths; (i) the packet is directly transmitted to c, and (ii) the packet is first transmitted to b and next transmitted to c. The second path might be an

Table 1. $\bar{\lambda}$ and $\bar{\kappa}$ for the BA, WS, and KE models. In these simulations, we constructed 30 different networks for each model, and averaged the results.

	BA	WS ($r_p = 0.7$)	KE ($\mu = 0.2$)
$\bar{\lambda}$	0.0941	0.038	0.491
$\bar{\kappa}$	2.741	2.373	3.423

https://doi.org/10.1371/journal.pone.0283472.t001

unnecessary detour depending on the distribution of stored packets. The SP and SPr methods transmit the packets using the first path because these methods only consider the shortest distance. However, the memory method detours packets using the memory information obtained by the transmitting node. Thus, the packet might be unnecessarily detoured even if the packet should be transmitted directly to node *c* because of the unsuitable strength of the memory information. The performance of the memory method for the KE model was then degraded.

To clarify how many networks have triangular connections and how far apart are the nodes in the networks, we next measured the clustering coefficient ($\bar{\lambda}$) and the average shortest hop distance between nodes ($\bar{\kappa}$) [31]. The clustering coefficient ($\bar{\lambda}$) is defined as follows:

$$\bar{\lambda} = \frac{1}{|V|}\sum_{i \in V} \lambda_i, \tag{14}$$

$$\lambda_i = \frac{2}{k_i(k_i - 1)}\sum_{j \in V}\sum_{s \in V} a_{ij}a_{js}a_{si}, \ \ i \in V, \tag{15}$$

$$a_{ij} = \begin{cases} 1 & (\text{if node } i \text{ and } j \text{ are connected}), \\ 0 & (\text{otherwise}). \end{cases} \tag{16}$$

In Eq (15), $a_{ij}a_{js}a_{si} = 1$ if the nodes *i*, *j*, and *s* are triangularly connected. Namely, Eq (15) averaged the number of triangular connections among the nodes *i*, *j*, and *s*. Therefore, the clustering coefficient ($\bar{\lambda}$) in Eq (14) measured an average ratio of triangular connections in the networks. If $\bar{\lambda}$ increases, the network has many triangular connections. The average shortest hop distance between nodes ($\bar{\kappa}$) is defined as follows:

$$\bar{\kappa} = \frac{1}{|V|(|V| - 1)}\sum_{i \in V}\sum_{j \in V, i \neq j} d_{ij}. \tag{17}$$

If $\bar{\kappa}$ takes large values, the networks have long hop distances between nodes.

Table 1 shows the clustering coefficients ($\bar{\lambda}$) and the average shortest hop distance between nodes ($\bar{\kappa}$) averaged over 30 different networks generated by the BA, WS, and KE models. In Table 1, the KE model has the largest $\bar{\lambda}$ among all the models. In addition, the KE model has larger $\bar{\kappa}$ than those of the BA and WS models. From Fig 3 and Table 1, the memory method results in low *A* values for the networks with many triangular connections and long distances between nodes. If the network has large $\bar{\kappa}$, the packets need many hops to be transmitted to the destinations. Thus, the point where the value of *A* starts decreasing from 100% of the memory method for the KE model is lower than that for the SF and WS models because $\bar{\kappa}$ of the KE model is larger than the other models. Additionally, the performance of the memory method for the KE model is further degraded because of its triangular connections.

Routing method that adaptively adjusts the weight of memory information

From the results in previous section, the memory method showed a low transmission completion rate of packets for the KE model. To improve this method, we first investigate transmitting probabilities between any connected nodes. The transmitting probability of packets from node i and its adjacent node j, $p_{ij}(I)$, is defined as:

$$p_{ij}(I) = \frac{\sum_{t=1}^{I} l_{ij}(t)}{\sum_{t=1}^{I}\sum_{s\in\partial_i} l_{is}(t)}, \qquad i \in V, j \in \partial_i, \tag{18}$$

where $l_{ij}(t)$ is the total number of packet transmissions from node i to its adjacent node j at iteration t.

Fig 4 illustrates the transmitting probabilities of the typical four edges by the memory method for the BA model. In these simulations, we randomly selected four typical edges from different nodes in the networks. Although transmitting probabilities converged to different values, each probability showed almost a constant value as the iterations progressed. These results lead to the following assumption: even if we incorporate the fixed transmitting probabilities into the memory information, instead of Eq (8), the routing method would show a similar performance as that of the memory method. To confirm this, we modified the memory information $\zeta_{ij,\tau}$ by using the transmitting probabilities. The modified memory information, $\zeta'_{ij,\tau}$, is defined as follows:

$$\zeta'_{ij,\tau} = \begin{cases} \alpha \sum_{s=0}^{\tau} \gamma^s x_{ij}(\tau - s), & (t \le L), \\ \dfrac{(k_{\max} + k_{\min})}{2|V|} p_{ij}(L), & (L < t), \end{cases} \tag{19}$$

where L is a period of iterations for calculating the transmitting probability, and $k_{\min}$ and $k_{\max}$ are the minimum and maximum degrees in the networks, respectively. A coefficient $(k_{\max} + k_{\min})/(2|V|)$ expresses a median of the degree and balances the weight of modified memory information. We heuristically found this weight in preliminary numerical simulations. In Eq (19), $p_{ij}(L)$, defined in Eq (18), is calculated when the iteration t is L. Further, we defined an

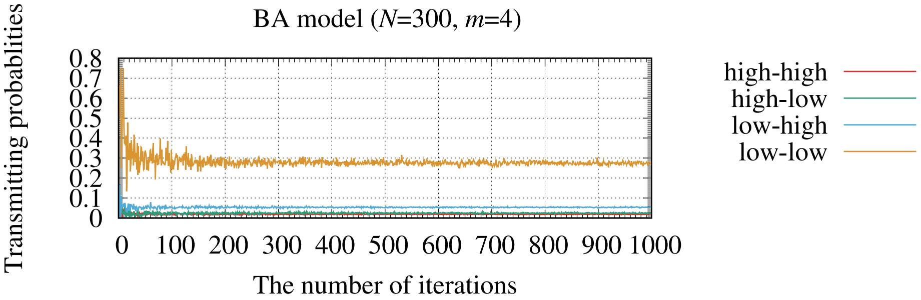

Fig 4. Typical examples of transmitting probabilities of four selected edges by the memory method for the BA model. In these simulations, we randomly selected four typical edges from different nodes in the networks. In this figure, "high-high," "high-low," "low-high," and "low-low" are transmitting probabilities that connect high-degree nodes, high and low-degree nodes, low and high-degree nodes, and low-degree nodes, respectively.

https://doi.org/10.1371/journal.pone.0283472.g004

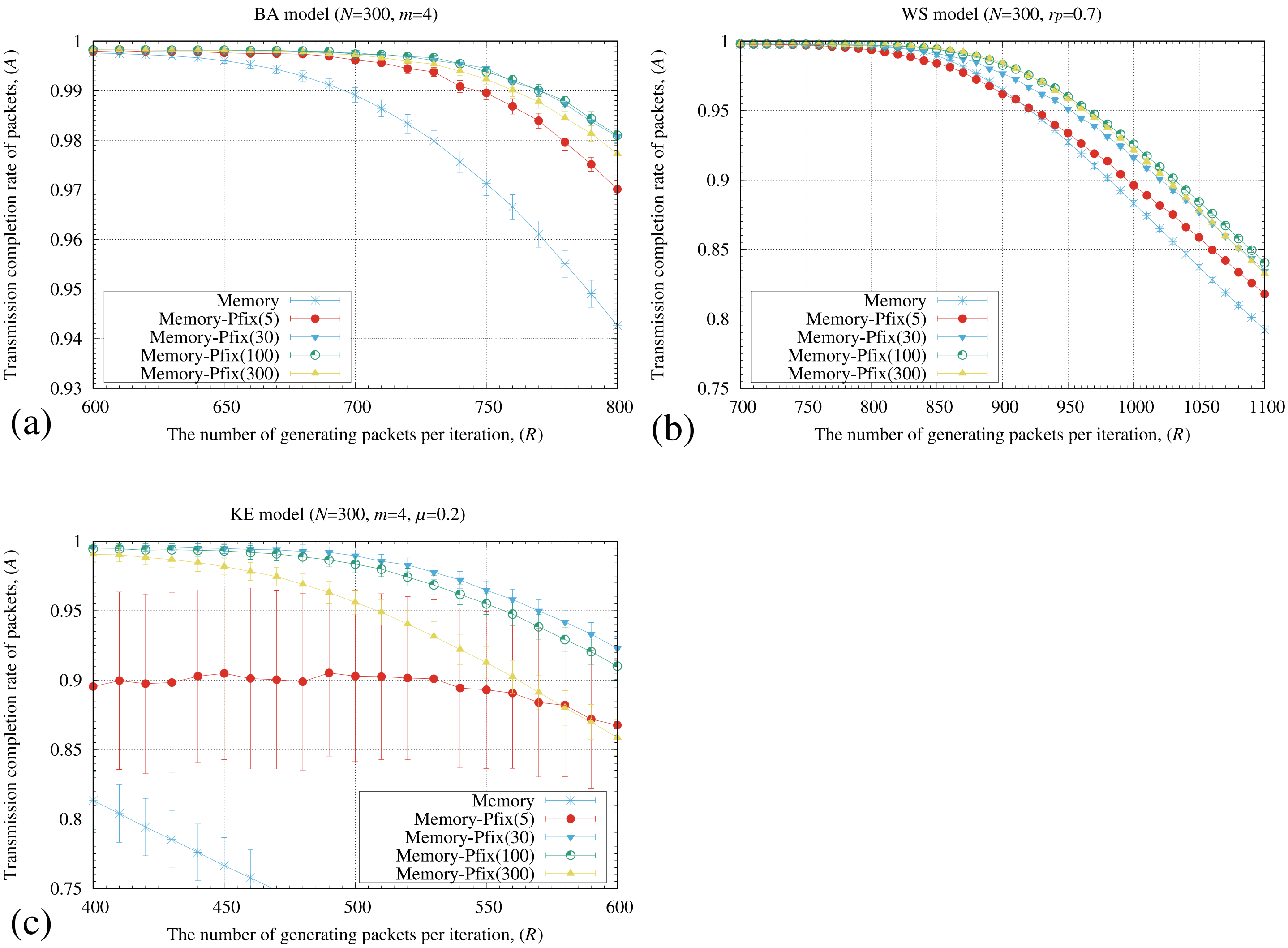

Fig 5. Relationship between the number of generating packets at each iteration (R) and a transmission completion rate of packets (A) by the original memory method and the memory-pfix method for the (a) BA, (b) WS, and (c) KE models. The numbers in the parentheses in the memory-pfix method indicate the periods of iterations for calculating the transmitting probability (L). In these figures, the standard deviation of each method is plotted as the error bars.

https://doi.org/10.1371/journal.pone.0283472.g005

evaluation function for this routing method as follows:

$$\min_{j\in\partial_i} y'_{ij,\tau}, \qquad i \in V, \;\; 1 \leq \tau \leq \omega_{ij}, \tag{20}$$

where $y'_{ij,\tau} = \xi_{ij} + \zeta'_{ij,\tau}$.

Eqs (8), (10), (18), (19) and (20) compose the memory method with fixed transmitting probabilities (hereafter, we call this method *memory-pfix*. The memory-pfix method transmits the packets using the static paths determined by the distance and modified memory information. These paths might be better than the shortest paths by using the SP and SPr methods. We then compared the transmission completion rate of packets of the original memory method [24] with the memory-pfix method.

Fig 5 shows the transmission completion rate of packets (A) by the original memory and memory-pfix methods for the BA, WS, and KE models. We constructed 30 different networks generated by the BA, WS, and KE models and averaged the results. We used different periods

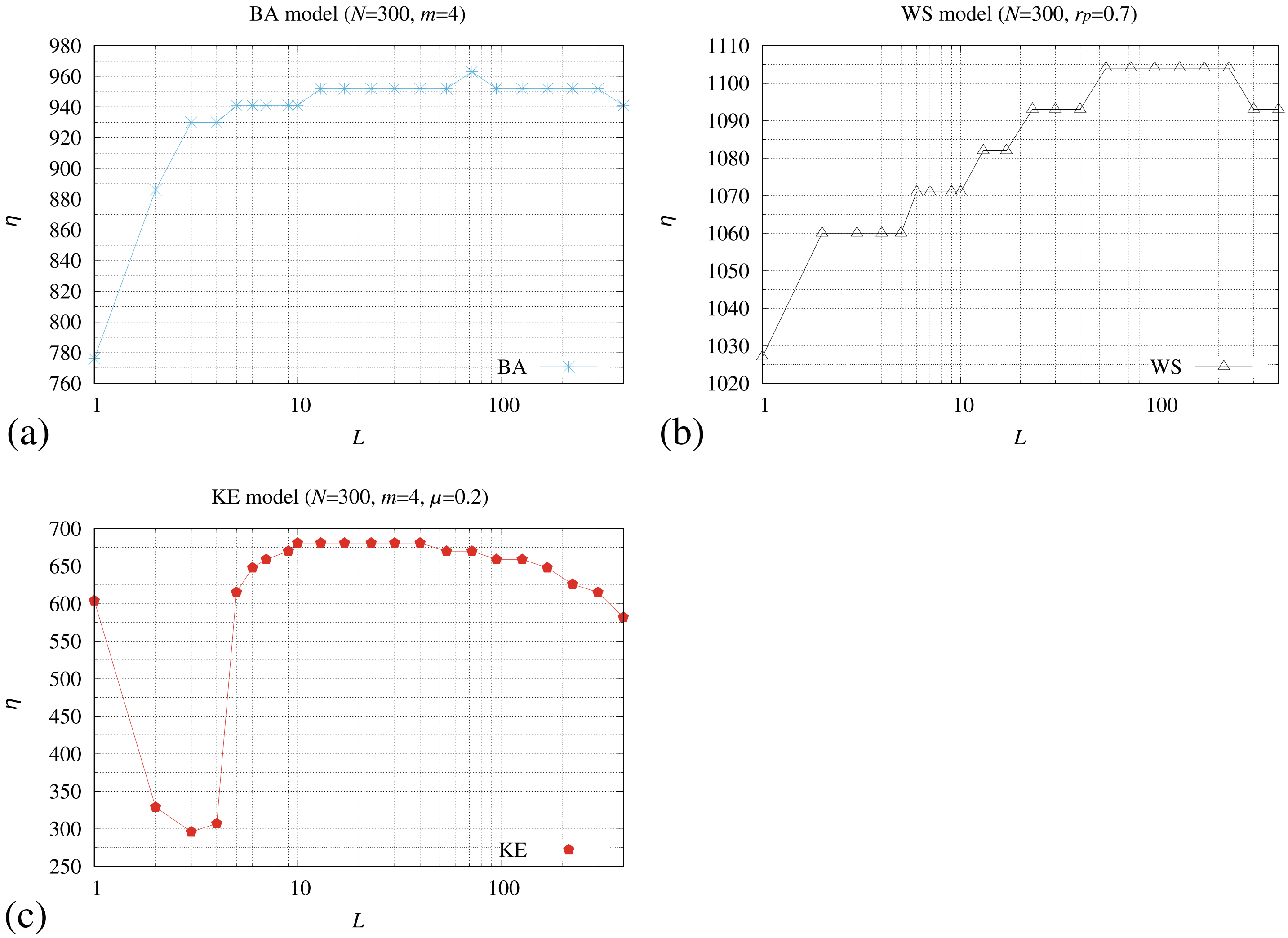

Fig 6. Relationship between a period of iterations for calculating the transmitting probability (L) in Eq (19) and the maximum value of R, where A in Eq (7) is lower than or equal to 0.9 of the memory-pfix method (η) for the (a) BA, (b) WS, and (c) KE models.

https://doi.org/10.1371/journal.pone.0283472.g006

of iterations for calculating the transmitting probability (L) and evaluated the method. The numbers in parentheses in the figure legends denote the L values. In Fig 5, the memory-pfix method showed higher A values compared to those of the original memory method for all the models.

Fig 6 shows the maximum value of R, where A in Eq (7) is lower than or equal to 0.9 in the memory-pfix method (η) as a function of a period of iterations for calculating the transmitting probability (L) in Eq (19) for the BA, WS, and KE models. We constructed 30 networks for each model and calculated the averaged value of η. In these figures, the values of η are high when $L > 10$ for BA model (Fig 6(a)), $60 < L < 110$ for WS model (Fig 6(b)), and $10 < L < 100$ for KE model (Fig 6(c)).

One of the strategies to further enhance routing performance is to adjust $p_{ij}(L)$ depending on the states of adjacent nodes. To realize this, we changed $\zeta'_{ij,\tau}$ in Eq (19) as follows:

$$\phi_{ij,\tau}(t) = \begin{cases} \alpha \sum_{s=0}^{\tau} \gamma^s x_{ij}(\tau - s), & (t \leq L), \\ \tilde{p}_{ij}(t), & (L < t), \end{cases} \tag{21}$$

where

$$\tilde{p}_{ij}(t) = \frac{p'_{ij}(t)}{\sum\limits_{s \in \partial_i} p'_{is}(t)}, \tag{22}$$

$$p'_{ij}(t) = \begin{cases} \dfrac{(k_{max} + k_{min})}{2|V|} p'_{ij}(t-1), & (\text{if } q_j(t)/\tilde{C}_j < \theta), \\ \sqrt{\dfrac{k_j}{k_{max}}}, & (\text{otherwise}), \end{cases} \tag{23}$$

where we defined $p'_{ij}(L) = p_{ij}(L)$. In Eq (23), $q_j(t)$ is the number of stored packets of node j at the tth iteration, $\tilde{C}_j$ is the modified transmission performance defined by Eq (6), and θ is a threshold value that determines whether the adjacent node j is congested. Eq (22) works to change the probability of transmitting packets to each of the adjacent nodes adaptively; when the number of stored packets in each adjacent node occupied over $100 \times \theta\%$ of its buffer size, $p'_{ij}(t)$ is recalculated by Eqs (22) and (23), thereby changing transmitting probabilities so that the packets can be transmitted to adjacent nodes with plenty of free space in their buffer after the iteration passes L in Eq (21). We defined that a node is in the free-flow state when $q_j(t)/\tilde{C}_j < \theta$ and in the congested state when $q_j(t)/\tilde{C}_j \geq \theta$. If the adjacent node is congested, the weight of memory information is set to $\sqrt{k_j/k_{max}}$.

In the networks, the nodes that transmit a large number of packets are divided into the following two types; (i) a node that has a small degree but has a large node betweenness centralities, and (ii) a node that has both a large degree and node betweenness centralities. We call the second type of node a *hub* node. If the first type of node is congested, the transmissions of packets to this node can be avoided by changing $\tilde{p}_{ij}(t)$ in Eq (23) to $\sqrt{k_j/k_{max}}$. However, our method continues to transmit the packets to the hub nodes even if these nodes are congested because if the hub nodes are congested, its $\tilde{p}_{ij}(t)$ becomes $\sqrt{k_j/k_{max}} \simeq 1$, and this achieves $\{(k_{max} + k_{min})/(2|V|)\}p'_{ij}(t-1) < \sqrt{k_j/k_{max}}$. This implies that our method continues to use the hub node for transmitting packets independent of the amounts of packets in the networks because the hub nodes have high storage and transmission performance for routing the packets. In addition, our method tries to avoid the transmissions of the packets to the nodes that have small degree and large betweenness centralities. We expect that this strategy for avoiding the congestion of packets is effective for routing the packets in the networks.

Moreover, we redefined an evaluation function for our proposed method as follows:

$$\min_{j\in\partial_i} z_{ij,\tau}(t), \qquad i \in V,\ 1 \le \tau \le \omega_{ij}, \tag{24}$$

where $z_{ij,\tau}(t) = \xi_{ij} + \phi_{ij,\tau}(t)$.

By using Eqs (8), (10), (21), (23) and (24), we finally propose a routing method that automatically adjusts the weight of memory information based on the memorized routing history. In our method, recalculations of distance information are not necessary once the networks are constructed. In addition, our method locally and less frequently exchanges information between the connected nodes; a node transmits a signal (0 or 1) to its adjacent nodes only if the node is congested (see S2 File for details). Hereafter, we call our proposed method *memory-pauto*.

Before starting the routing of packets, we prepared the routing tables at each node by calculating the shortest hop distance between any pairs of nodes. Because the SP method transmits the packets using the fixed transmitting paths, the method only refers to an index of the next adjacent node when routing the packets. The SP method then needs $O(1)$ calculation cost at each node. However, the SPr method randomly selects an adjacent node if multiple adjacent nodes have the same shortest hop distance to the destination. If all the adjacent nodes of a transmitting node have the same shortest hop distances, and this transmitting node has the maximum degree, defined by $k_{\max}$, the SPr method needs $O(k_{\max})$ to determine the next transmitting node in the worst case. The ER method uses information of the shortest hop distances and the information of stored packets at adjacent nodes. This needs $O(k_i)$ at each node. Thus, the ER method requires $O(k_{\max})$ to determine the transmitting node in the worst case. The calculation costs by the memory-pauto method change depending on the values of L that determines the transmitting probabilities defined by Eq (21). When $t < L$, The memory-pauto uses the distance information in Eq (8) and modified memory information in Eq (21). At this time, each node needs $O(1)$ to obtain the memory information between connected pairs of nodes because Eq (21) is the same as Eq (9). In addition, the calculation costs for obtaining the distance information are the same as the ones used in the ER method. Namely, each node in the memory-pauto method requires $O(k_i)$ when $t < L$. When $L \le t$, recalculations of transmitting probabilities by using Eqs (22) and (23) are required and these calculation costs become $O(k_i)$ at each node. Please note that this recalculation is required only if the state of the node changes. By summarizing these, in the worst case, the memory-pauto method requires $O(k_{\max})$ to determine the next transmitting node. When $t = L$, the memory-pauto calculates the transmitting probability of every edge by using Eq (22). This requires $O(Lk_i)$ at each node. However, this calculation only performed when $t = L$, and we omitted this from the calculation costs. All the routing methods require the distance information to transmit the packets. To obtain this information, we used the Dijkstra algorithm [30]. Generally, the Dijkstra algorithm needs $O(|V|^2)$, but it changes $O(|E| + |V| \log |V|)$ using a Fibonacci heap. Other methods such as the bread-first search or A* algorithm are also applicable to calculate the shortest distance between any pairs of nodes. However, these algorithms only performed when the networks were constructed. Therefore, we omitted this from the calculation costs for each routing method.

Numerical experiments

The routing performance of the memory-pauto method mainly depends on L and θ in Eqs (21)–(23). L determines the period of iterations for calculating the transmitting probability, and θ determines whether the node is congested. We first evaluated the performance dependencies by these parameters.

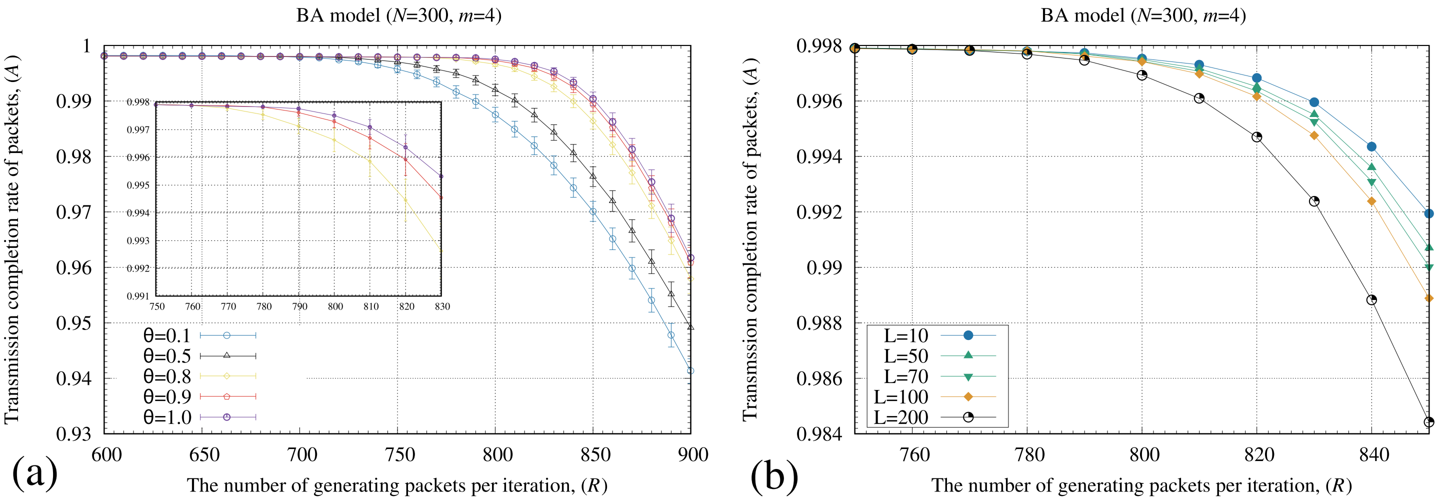

Fig 7. Relationships between the number of generating packets at each iteration (R) and a transmission completion rate of packets (A) for different L and θ values applied for the memory-pauto method. In these figures, the standard deviation of each method is plotted as error bars. In these numerical experiments, we used the BA model.

https://doi.org/10.1371/journal.pone.0283472.g007

Fig 7 shows the transmission completion rate (A) by incorporating different θ and L values to the memory-pauto method. In Fig 7(a), the memory-pauto method maintains the highest A value if θ equals 1. In Fig 7(b), the memory-pauto method maintains the highest A value if L equals 10.

Fig 8(a)–8(c) shows the maximum value of R, where A in Eq (7) is less than or equal to 0.9 of the memory-pfix method (η) when increasing the value of threshold (θ) that determines whether the adjacent nodes are congested. Fig 8(d)–8(f) shows the maximum value of R, where A in Eq (7) is less than or equal to 0.9 of the memory-pfix method (η) hen increasing the period of iterations for calculating the transmitting probability (L) in Eq (21). In these simulations, we set L to 5 in Fig 8(a)–8(c), and $\theta = 1.0$ in Fig 8(d)–8(f). In Fig 8(a)–8(c), the memory-pauto shows the largest values of η if $\theta > 0.65$ for BA, $\theta > 0.2$ for WS, and $\theta > 0.75$ for KE models. In Fig 8(d)–8(f), the memory-pauto shows the largest values of η when $L \leq 40$ for BA, $2 \leq L \leq 7$ for WS, and $L \leq 40$ for KE models.

These results indicate that the memory-pauto method works well if each node holds as many packets as possible within a range of its modified transmission performance defined by Eq (6) because the method shows large η values if θ increases. Moreover, the memory-pauto method needs short periods to determine the transmitting probabilities because η takes large values if L decreases. Comparing with Fig 6, the memory-pauto method requires short periods for determining the transmitting probability because $p'_{ij}(t)$ in Eq (23) changes depending on the states of adjacent nodes. Based on Fig 8, we set θ and L of the memory-pauto method to 1.0 and 5 for all the models, respectively.

We next evaluated the routing performance of the memory-pauto method for the BA, WS, and KE models. These numerical experiments evaluated the transmission completion rate of packets (A) in Eq (7). We also evaluated the average hop (H) and average arrival time (U) of arriving packets. The H and U are defined as follows:

1. Average hops of arriving packets H:

$$H = \frac{1}{|v|}\sum_{m\in v} h_m, \tag{25}$$

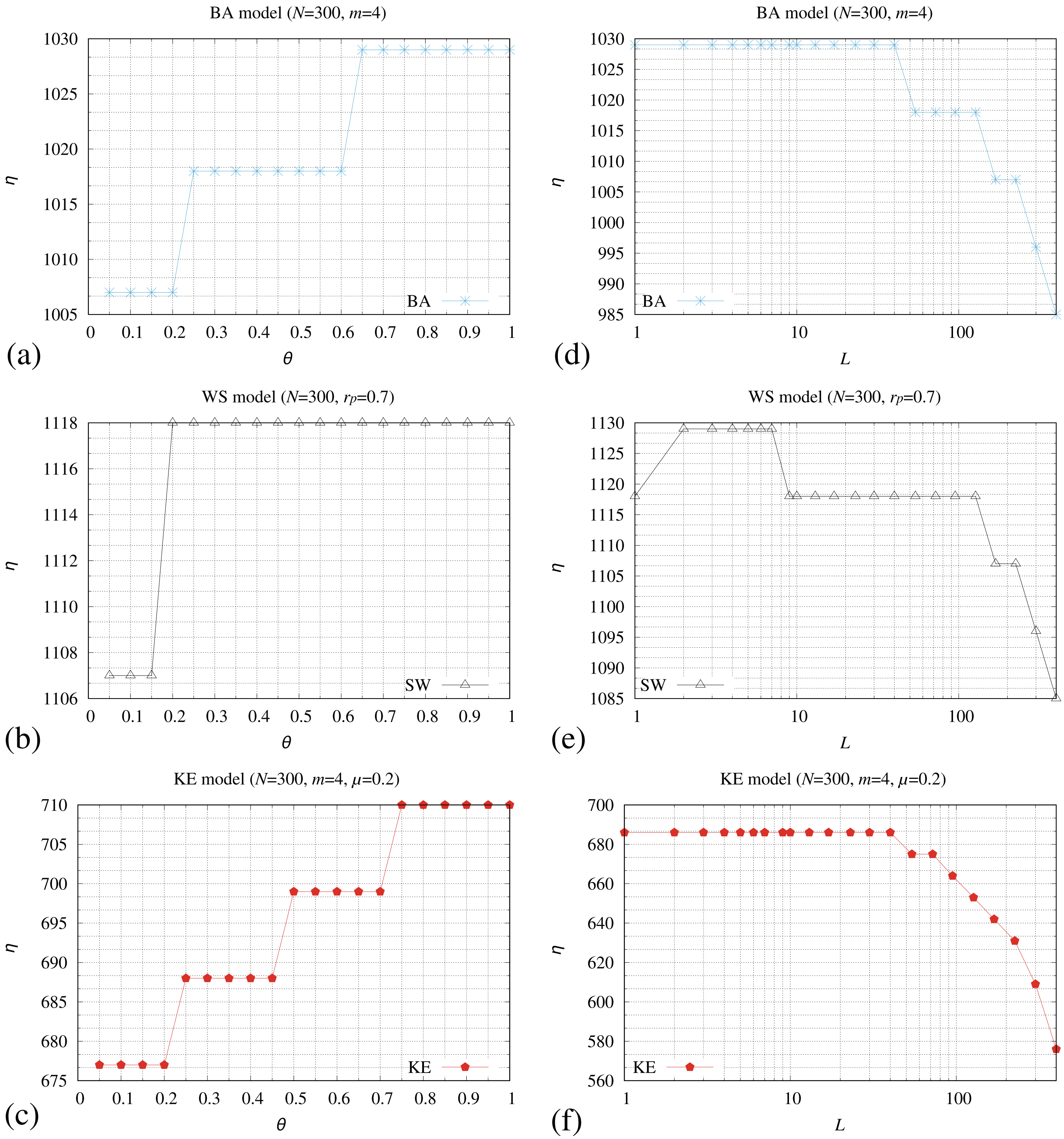

Fig 8. Relationships between a threshold (θ) that determines whether the adjacent nodes are congested and the maximum value of R, where $A \leq 0.9$ (η) of the memory-pfix method for the (a) BA, (b) WS, and (c) KE models, and relationships between a period of iterations for calculating the transmitting probability (L) and η for the (d) BA, (e) WS, and (f) KE models. In these simulations, we set L to 5 in (a), (b), and (c), and $\theta = 1.0$ in (d), (e), and (f).

https://doi.org/10.1371/journal.pone.0283472.g008

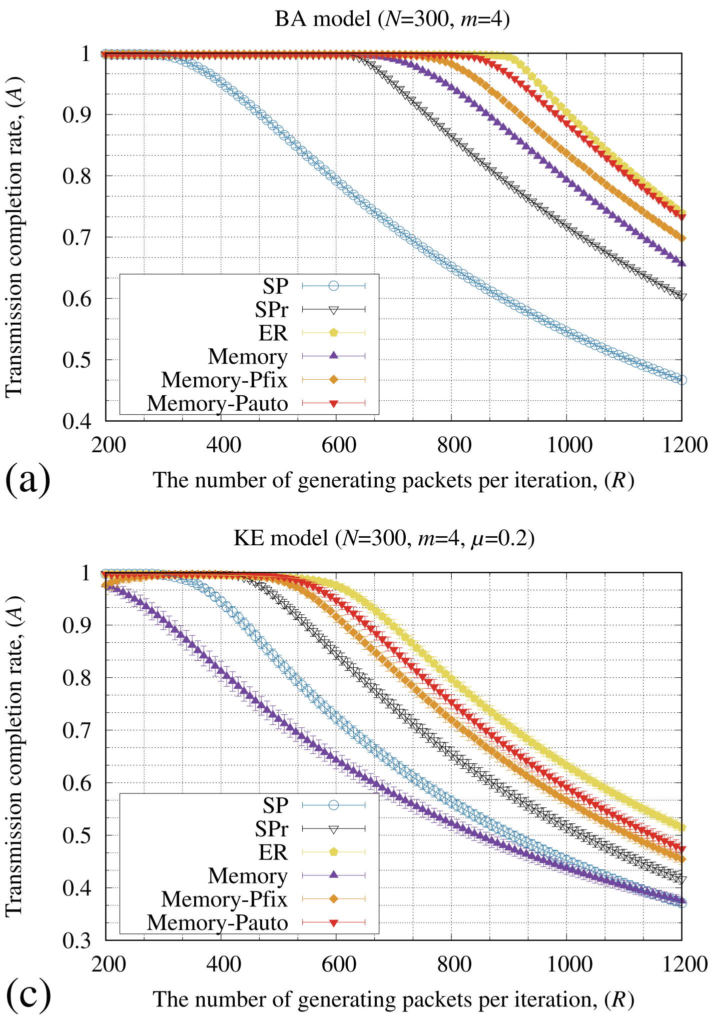

Fig 9. Relationships between the number of generating packets at each iteration (*R*) and a transmission completion rate of packets (*A*) of the SP, SPr, ER, memory, memory-pfix, and memory-pauto for the (a) BA, (b) WS, and (c) KE models. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g009

where v is a set of arriving packets, $|v|$ is the number of elements in v, and h_m is the number of hops from the source to the destination of packet m.

2. Average arrival time of arriving packets U:

$$U = \frac{1}{|v|}\sum_{m\in v} u_m, \tag{26}$$

where u_m is the number of iterations required to transmit packet m from the source to the destination.

In these simulations, we set L in Eq (18) for the memory-pfix to 70 for BA, 100 for WS, and 20 for KE models based on Fig 6. Other experimental conditions were the same as the ones described in previous sections. We constructed 30 different networks and averaged the results.

Fig 9 shows the transmission completion rate of packets (A) when the number of generating packets at each iteration (R) increases by the SP, SPr, ER, memory, memory-pfix, and memory-pauto methods for the BA, WS, and KE models. In Fig 9(a)–9(c), the memory-pauto

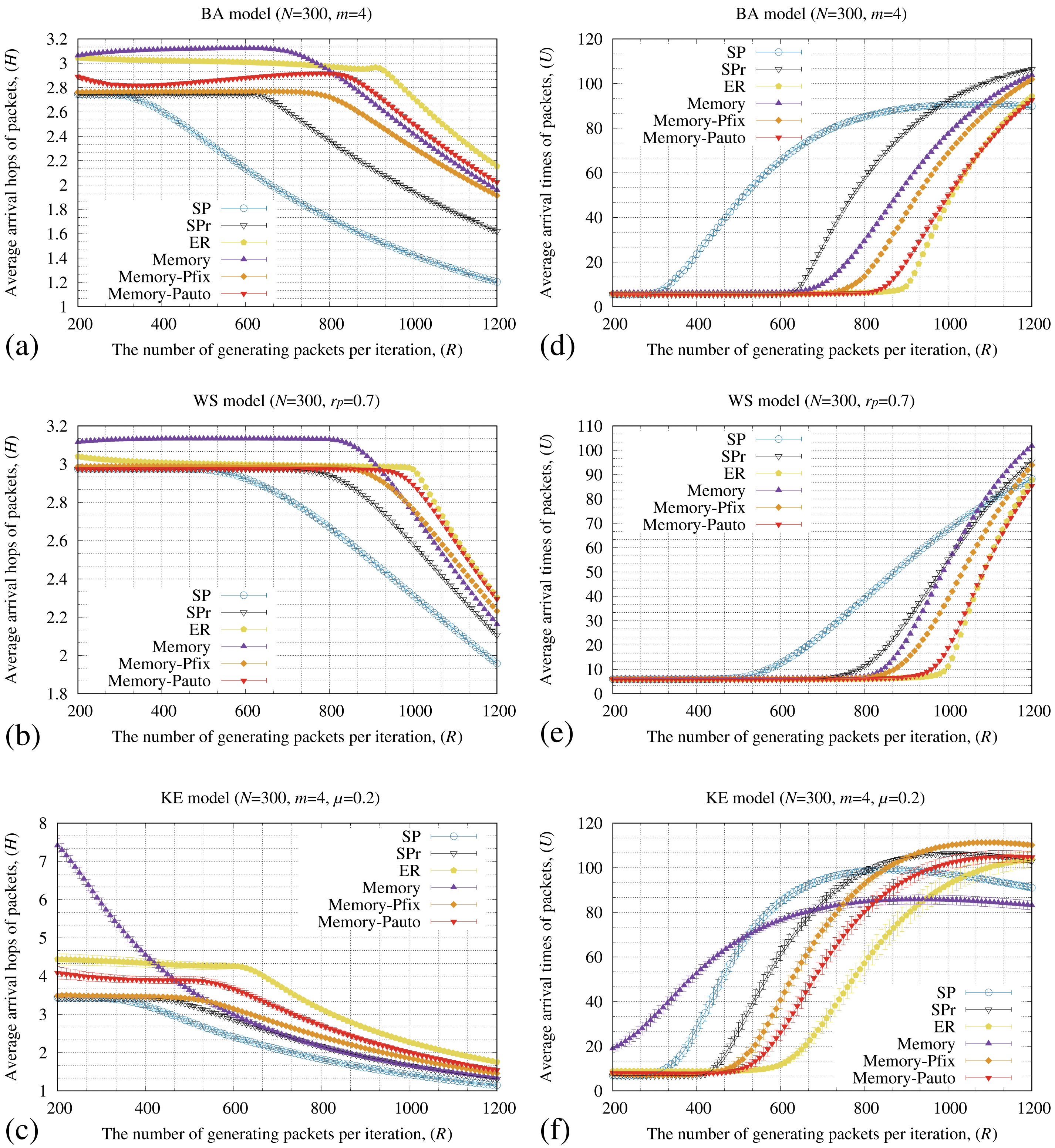

Fig 10. Relationships between the number of generating packets at each iteration (*R*) and the average hop of arriving packets (*H*) of the SP, SPr, ER, memory, memory-pfix, and memory-pauto methods for the (a) BA, (b) WS, and (c) KE models, and the average arrival times of arriving packets (*U*) for the (d) BA, (e) WS, and (f) KE models. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g010

method maintains a higher A than the SP, SPr, and memory-pfix methods for all the models. In particular, the maximum value of R, where A starts to decrease from 100%, is improved by approximately 170% by our proposed method for the BA model compared to the SP method and 25% compared to the memory method. In addition, the ER and memory-pauto methods show similar A values for the WS model. We also evaluated these routing methods for BA, WS and KE models with small degrees (see S3 Fig in S3 File for details).

Fig 10 shows the average arrival hops of arriving packets (H) and their average arrival time (U) by the SP, SPr, ER, memory, memory-pfix, and memory-pauto methods for the BA, WS, and KE models. In Fig 10(a)–10(c), the ER and memory-pauto methods show the longer H value as R increases compared to those for the the SP, SPr, memory, and memory-pfix methods. Moreover, the values of H for all the routing methods decreases as R increases because only the packets whose sources and destinations are directly connected arrive at their destinations when R increases enough. In Fig 10(d)–10(f), the memory-pauto exhibited long H, short U, and high A values. These results demonstrate that the memory-pauto method effectively

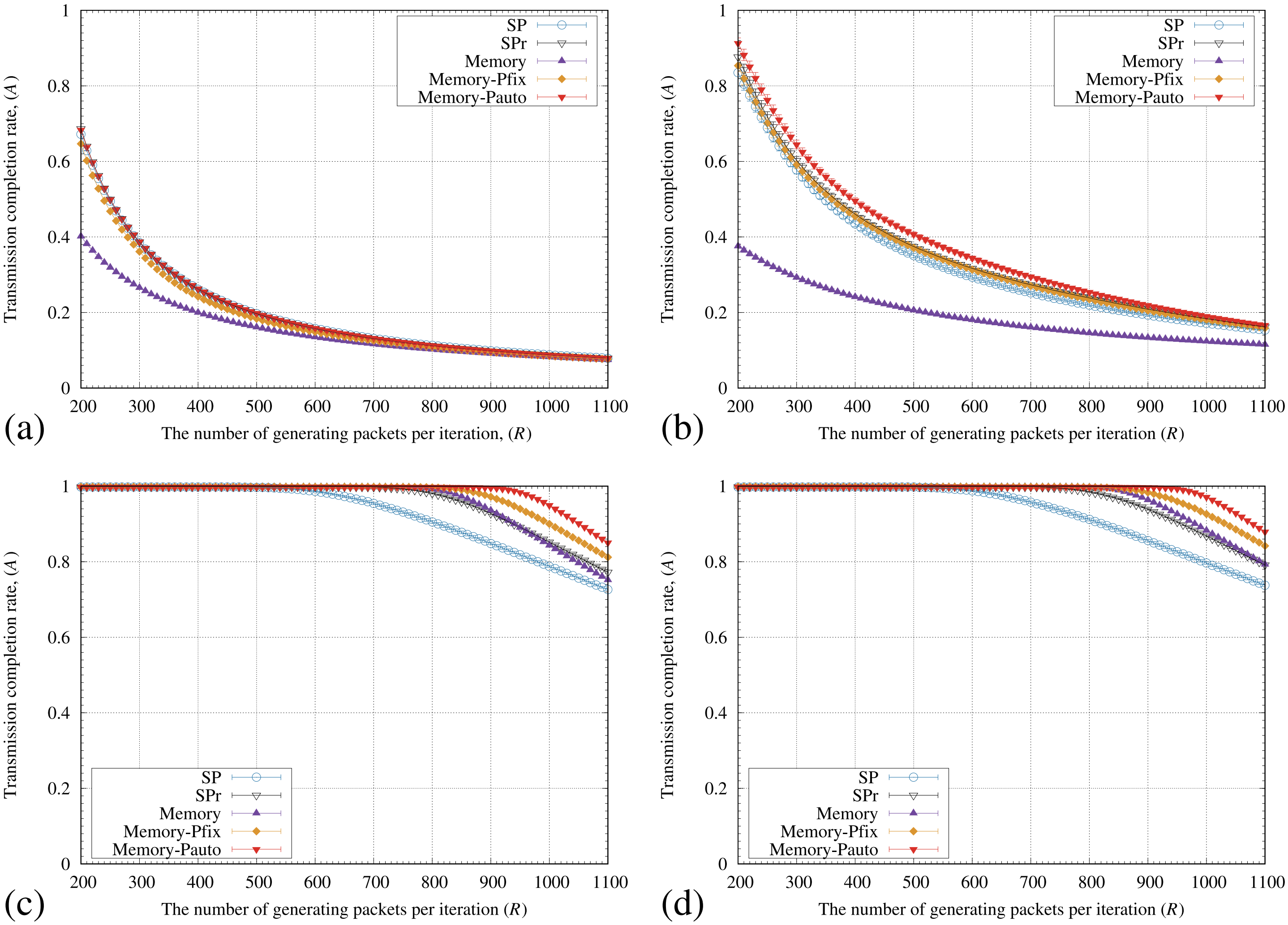

Fig 11. Relationships between the number of generating packets at each iteration (R) and the transmission completion rate of arriving packets (A) of the SP, SPr, memory-pfix, and memory-pauto methods for the WS model with (a) $r_p = 0$, (b) $r_p = 0.01$, (c) $r_p = 0.5$, and (d) $r_p = 1.0$. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g011

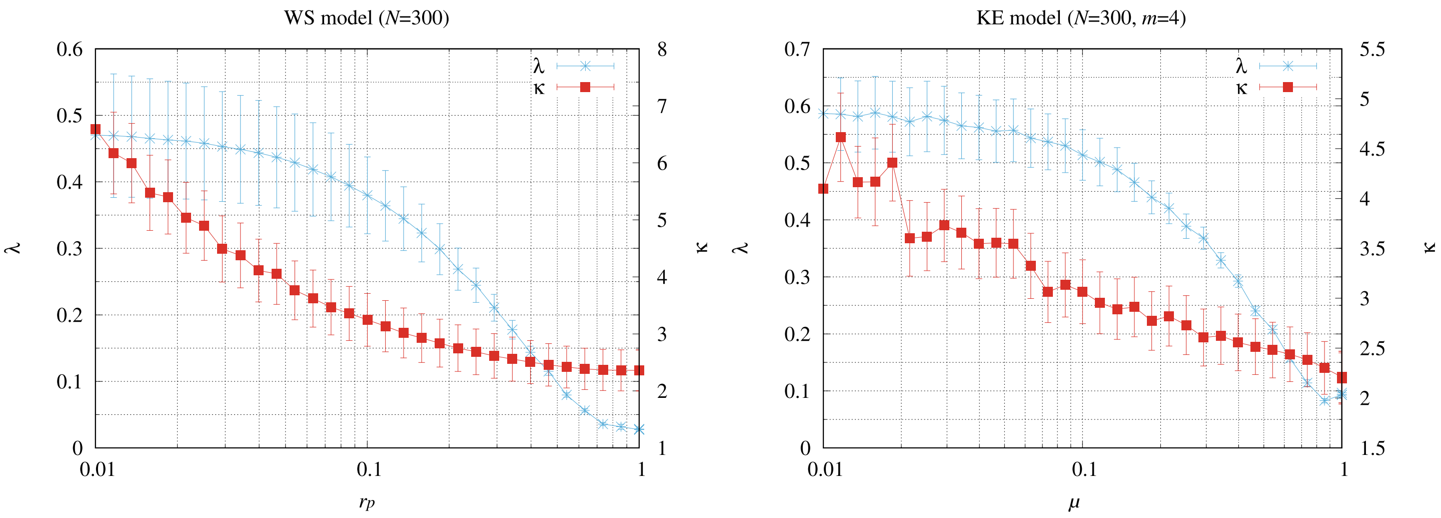

Fig 12. Relationships between the clustering coefficient ($\bar{\lambda}$) and the average shortest hop distance between nodes ($\bar{\kappa}$) if we change (a) the rewiring probability (r_p) for the WS model, and (b) the probability (μ) that adjusts the clustering coefficient for the KE model. In these simulations, we constructed 30 different networks for each model, and averaged the results.

https://doi.org/10.1371/journal.pone.0283472.g012

diversifies the transmitting routes for packets, which increases the hops for the packets to the destinations.

Moreover, the total iterations of the packets to be transmitted from the sources to destinations are shortened by the effective transmissions. As a result, the proposed method shows the highest transmission completion rate even for large volumes of traffic flow in the communication networks.

In the above numerical experiments, we fixed the rewiring probability (r_p) for the WS model to 0.7 and the probability (μ) that adjusts the clustering coefficient for the KE model to 0.2. However, the routing performance changes are based on the topologies of the networks. Thus, we clarified the performance dependencies against these probabilities.

Fig 11 illustrates the transmission completion rate of packets (A) of SP, SPr, memory, memory-pfix, and memory-pauto methods for various values of the rewiring probability (r_p). In Fig 11, the point where A starts to decrease from 100% by all the methods increases as r_p increases. In addition,A by the memory method is lower than those for the SP and SPr methods when the r_p values are 0 and 0.01. In contrast, the memory-pfix and memory-pauto methods maintain the higher A values than the SP and SPr methods for the WS model (Fig 11(c) and 11(d)).

Fig 12 shows a relationship between the clustering coefficient ($\bar{\lambda}$) in Eq (14) and the average shortest hop distance between nodes ($\bar{\kappa}$) in Eq (17) if we changed the rewiring probability (r_p) for the WS model (Fig 12(a)) and the probability (μ) that adjusts the clustering coefficient for the KE model (Fig 12(b)). In these simulations, we constructed 30 different networks for each model and averaged the results. In Fig 12(a), the WS model has large values of $\bar{\lambda}$ and $\bar{\kappa}$ when r_p is set to 0 and 0.01. In these cases, the networks have many triangular connections and long distances between the nodes. However, the WS model has the small values of $\bar{\lambda}$ and $\bar{\kappa}$ when r_p is set to 0.5 and 1. In these networks, the packets need a few hops transmitted to their destinations. In addition, the networks have a small number of triangular connections. The routing performance of all the methods is then improved as r_p increases (Fig 11(c) and 11(d)).

Fig 13 illustrates the average arrival rate of packets (A) for various values of μ in the KE model. In Fig 13, the memory-pauto method shows the highest A value except for $\mu = 0$. In addition, the maximum values of R, where A starts to decrease from 100% by the memory-

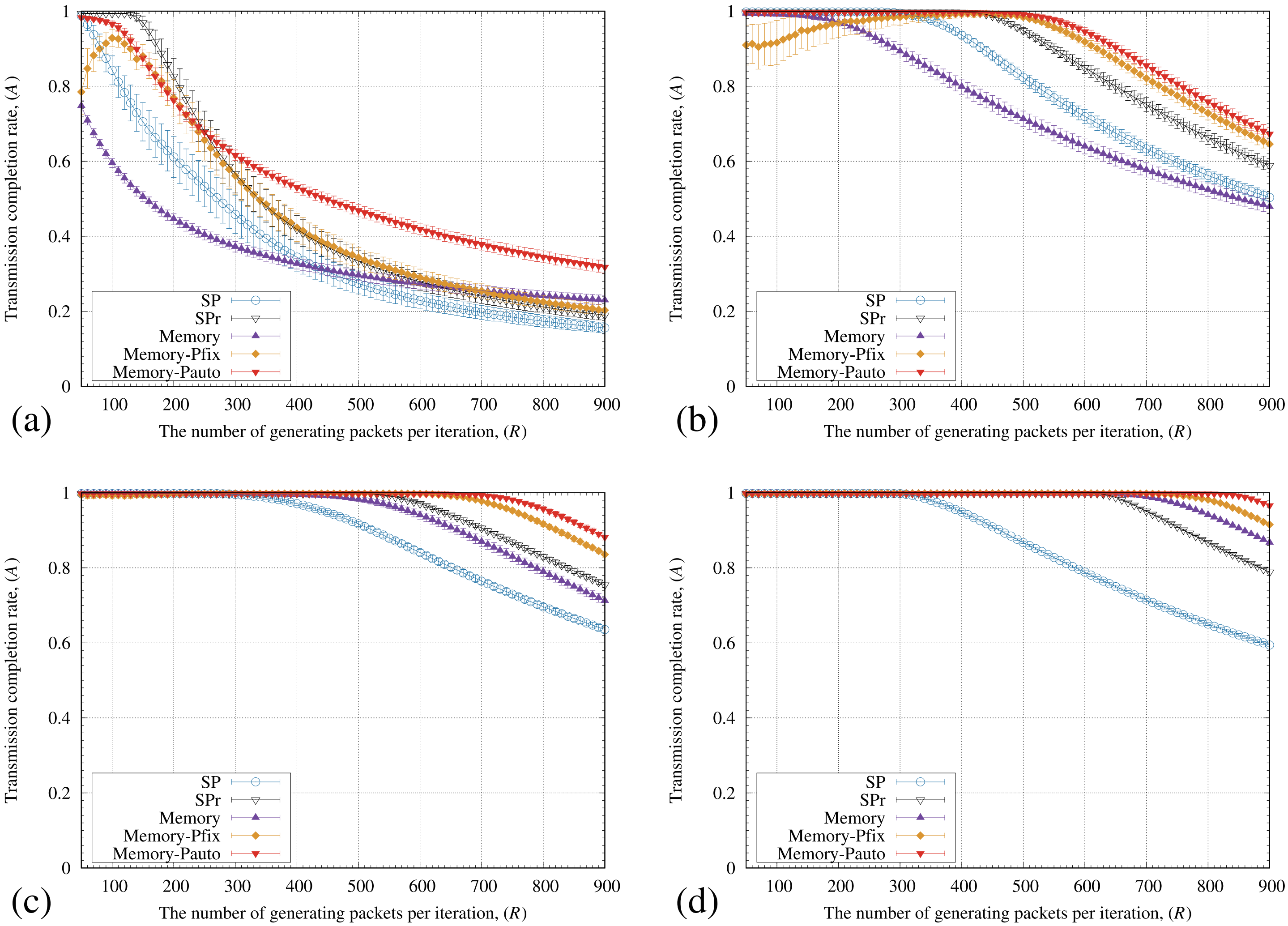

Fig 13. Relationships between the number of generating packets at each iteration (*R*) and a transmission completion rate of arriving packets (*A*) of the SP, SPr, memory, memory-pfix, and memory-pauto methods for the KE model with (a) $\mu = 0$, (b) $\mu = 0.2$, (c) $\mu = 0.5$, and (d) $\mu = 1$. In these figures, the standard deviation of each method is plotted as error bars.

https://doi.org/10.1371/journal.pone.0283472.g013

pauto method, increased as μ increased. In Fig 12(b), the KE model shows small values of $\bar{\lambda}$ and $\bar{\kappa}$ as μ increases. In addition, the KE model changed to the SF model if μ approached 1. Because the triangular connections decrease and the networks have short distances when $\bar{\lambda}$ and $\bar{\kappa}$ have small values,A values for all the methods are improved (Fig 13(c) and 13(d)). Figs 11 and 13 indicate that the memory-pauto method showed high A values for the networks with many triangular connections and long distances between nodes.

These results indicate that the memory-pauto method effectively diversifies the routes of transmitting packets to maintain a high transmission completion rate by adjusting adequate weights of memory information at each node, depending on the traffic volumes and topologies of the communication networks.

Conclusion

In this study, we improved the routing performance of the original routing strategy using memory information by automatically adjusting the weight of memory information based on states of adjacent nodes. First, we enhanced the routing performance of conventional communication network models by changing the packet transmission and storage performance based on their degree and on node betweenness centralities. Using the improved communication network models, the SP method, which is commonly applied in real communication networks was significantly improved. We also showed that the memory routing method maintained a high transmission completion rate compared to the SP method for the improved communication network models. However, the original memory routing method showed static transmitting probabilities as iteration moved, leading to the degradation of routing performance. Based on these consequences, we diversified the transmitting nodes of packets by changing weights of the memory information based on the transmitting probabilities between any connected nodes. Numerical simulations demonstrated that our proposed routing method improved transmission completion rate by approximately 170% compared to the SP method, and by 25% compared to the original routing method with the memory information for the communication networks with SF properties. Furthermore, our method displayed the highest transmission completion rate both for the networks with SF properties and also for those with highly clustered and with long distances between nodes. In future works, we plan to investigate the routing performance of our proposed method if the packet generating probability time-dependently changes depending on the local connectivity of networks.

Supporting information

S1 File. Performance evaluations of the original and modified ER methods.
(PDF)

S2 File. Performance evaluations with regard to the number of signals of the memory-pauto method.
(PDF)

S3 File. Performance evaluations of the network models with small and large degrees.
(PDF)

Author Contributions

Conceptualization: Takayuki Kimura, Yutaka Shimada.

Formal analysis: Takayuki Kimura.

Funding acquisition: Takayuki Kimura, Yutaka Shimada.

Investigation: Takayuki Kimura, Yutaka Shimada.

Methodology: Takayuki Kimura.

Project administration: Takayuki Kimura.

Resources: Takayuki Kimura.

Software: Takayuki Kimura.

Validation: Takayuki Kimura, Yutaka Shimada.

Visualization: Takayuki Kimura, Yutaka Shimada.

Writing – original draft: Takayuki Kimura.

Writing – review & editing: Takayuki Kimura, Yutaka Shimada.

References

1. Barabási AL, Albert R. Emergence of scaling in random networks. Science. 1999; 286:509–512. https://doi.org/10.1126/science.286.5439.509 PMID: 10521342
2. Echenique P, Gómez J, Moreno Y. Dynamics of jamming transitions in complex networks. Europhysics Letters. 2005; 71:325–331. https://doi.org/10.1209/epl/i2005-10080-8
3. Ling X, Hu MB, Jiang R, Wu QS. Global dynamic routing for scale-free networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2010; 81:016113. https://doi.org/10.1103/PhysRevE.81.016113 PMID: 20365438
4. Tang M, Liu Z, Liang X, Hui PM. Self-adjusting routing schemes for time-varying traffic in scale-free networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2009; 80:026114. https://doi.org/10.1103/PhysRevE.80.026114 PMID: 19792207
5. Wang WX, Wang BH, Yin CY, Xie YB, Zhou T. Traffic dynamics based on local routing protocol on a scale-free network. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2006; 73:026111. https://doi.org/10.1103/PhysRevE.73.026111 PMID: 16605402
6. Wang WX, Yin CY, Yan G, Wang BH. Integrating local static and dynamic information for routing traffic. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2006; 74:016101. https://doi.org/10.1103/PhysRevE.74.016101 PMID: 16907145
7. Yang HX, Wang WX, Xie YB, Lai YC, Wang BH. Transportation dynamics on networks of mobile agents. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2011; 83:016102. https://doi.org/10.1103/PhysRevE.83.016102 PMID: 21405739
8. Yang HX, Tang M. Adaptive routing strategy on networks of mobile nodes. Physica A: Statistical Mechanics and its Applications. 2014; 402:1–7. https://doi.org/10.1016/j.physa.2014.01.063
9. Gao X, Guo H, Chen Y, Tang Y, Wang C, Xu S, et al. Global Hybrid Routing for Scale-Free Networks. IEEE Access. 2019; 7:19782–19791. https://doi.org/10.1109/ACCESS.2019.2897255
10. Echagüe J, Cholvi V, Kowalski DR. Effective use of congestion in complex networks. Physica A: Statistical Mechanics and its Applications. 2018; 494:574–580. https://doi.org/10.1016/j.physa.2017.11.159
11. Echagüe J, Cholvi V, Fernández A. Factors affecting congestion-aware routing in complex networks. Physica A: Statistical Mechanics and its Applications. 2022; 587:126483. https://doi.org/10.1016/j.physa.2021.126483
12. Lin B, Chen B, Gao Y, Tse CK, Dong C, Miao L, et al. Advanced algorithms for local routing strategy on complex networks. PLoS ONE. 2016; 11. https://doi.org/10.1371/journal.pone.0163075
13. Zhang X, Zhou Z, Cheng D. Efficient path routing strategy for flows with multiple priorities on scale-free networks. PLoS ONE. 2017; 12. https://doi.org/10.1371/journal.pone.0172035 PMID: 28199382
14. Horiguchi T, Ishioka S. Routing control of packet tow using neural network. Physica A. 2001; 297:521–531. https://doi.org/10.1016/S0378-4371(01)00229-1
15. Horiguchi T, Hayashi K, Tretiakov A. Reinforcement learning for congestion-avoidance in packet flow. Physica A: Statistical Mechanics and its Applications. 2005; 349:329–348. https://doi.org/10.1016/j.physa.2004.10.015
16. Kimura T, Nakajima H, Ikeguchi T. A packet routing method for complex networks by a stochastic neural network. Physica A: Statistical Mechanics and its Applications. 2007; 376:658–672. https://doi.org/10.1016/j.physa.2006.10.061
17. Hou R, Chang Y, Yang L. Multi-constrained QoS routing based on PSO for named data networking. IET Communications. 2017; 11(8):1251–1255. https://doi.org/10.1049/iet-com.2016.0783
18. Hou R, Zhang L, Zheng Y, Chang Y, Li B, Huang T, et al. Service-differentiated QoS routing based on ant colony optimisation for named data networking. Peer-to-Peer Networking and Applications. 2019; 12(4):740–750. https://doi.org/10.1007/s12083-018-0669-6
19. Kimura T, Ikeguchi T. A new algorithm for packet routing problems using chaotic neurodynamics and its surrogate analysis. Neural Computing and Applications. 2007; 16:519–526. https://doi.org/10.1007/s00521-007-0099-5
20. Morita Y, Kimura T. An improved routing algorithm using chaotic neurodynamics for packet routing problems. Nonlinear Theory and Its Applications, IEICE. 2017; 9:95–106. https://doi.org/10.1587/nolta.9.95

21. Aihara K, Takabe T, Toyoda M. Chaotic neural networks. Physics Letters A. 1990; 144:333–340. https://doi.org/10.1016/0375-9601(90)90136-C

22. Ikeguchi T, Hasegawa M, Kimura T, Matsuura T, Aihara K. Theory and Applications of Chaotic Optimization Methods. In: Nedjah N, dos Santos Coelho L, Mariani VC, de Macedo Mourelle L, editors. Innovative Computing Methods and Their Applications to Engineering Problems. Springer; 2011. p. 131–161.

23. Kimura T. Congestion Avoidance on Networks Using Independent Memory Information. In: Visarath In AP Patrick Longhini, editor. Proceedings of the 5th International Conference on Applications in Nonlinear Dynamics. Understanding Complex Systems. Springer; 2019. p. 164–173.

24. Yoshida A, Shimada Y, Kimura T. Efficient routing strategy with transmission history information and its surrogate analysis. Physica A: Statistical Mechanics and its Applications. 2021; 566:125597. https://doi.org/10.1016/j.physa.2020.125597

25. Kantz H, Schreiber T. Nonlinear time series analysis. Cambridge Univercity Press, London; 2003.

26. Hu MB, Wang WX, Jiang R, Wu QS, Wu YH. Phase transition and hysteresis in scale-free network traffic. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2007; 75:036102. https://doi.org/10.1103/PhysRevE.75.036102 PMID: 17500754

27. Freeman LC. A Set of Measures of Centrality Based on Betweenness. Sociometry. 1977; 40:35–41. https://doi.org/10.2307/3033543

28. Yan G, Zhou T, Hu B, Fu ZQ, Wang BH. Efficient routing on complex networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2006; 73:046108. https://doi.org/10.1103/PhysRevE.73.046108 PMID: 16711879

29. Pu CL, Zhou SY, Wang K, Zhang YF, Pei WJ. Efficient and robust routing on scale-free networks. Physica A: Statistical Mechanics and its Applications. 2012; 391:866–871. https://doi.org/10.1016/j.physa.2011.08.044

30. Dijkstra EW. A note on two problems in connexion with graphs. Numerische Mathematik. 1959; 1 (1):269–271. https://doi.org/10.1007/BF01386390

31. Watts DJ, Strogatz SH, Watts DJ, Strogatz SH. Collective dynamics of “small-world” networks. Nature. 1998; 393:440–442. https://doi.org/10.1038/30918 PMID: 9623998

32. Klemm K, Eguíluz VM. Growing scale-free networks with small-world behavior. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 2002; 65:057102. https://doi.org/10.1103/PhysRevE.65.057102 PMID: 12059755