IEICE TRANS. INF. & SYST., VOL.E92–D, NO.10 OCTOBER 2009

PAPER

A Self-Adaptive Routing Protocol in Wireless LANs Based on Attractor Selection

Gen NISHIKAWA[†], ***Nonmember***, **Tomoko IZUMI**[††], **Fukuhito OOSHITA**[†], ***Members***, **Hirotsugu KAKUGAWA**[†], ***Nonmember***, ***and*** **Toshimitsu MASUZAWA**[†], ***Member***

SUMMARY Wireless LANs, which consist of access points and wireless stations, have widely spread in recent years. Routing in wireless LANs suffers the problem that each wireless station selects an access point and a wired path to its destination station. It is desired to design an adaptive routing protocol for wireless LANs since throughputs of communications are dynamically affected by selections of other wireless stations and external environmental changes. In this paper, we propose a routing protocol for wireless LANs based on *attractor selection*. Attractor selection is a biologically inspired approach, and it has high adaptability to dynamic environmental changes. By applying attractor selection, each wireless station can adaptively select its access point and wired path with high throughput against environmental changes. In addition, we design the protocol with a new technique: combination of multiple attractor selections. The technique is useful because it enables us to divide a problem into several simpler problems. To the best of our knowledge, our protocol is the first one designed around a combination of multiple attractor selections. We show the effectiveness and adaptability of our protocol by simulations.

key words: *attractor selection, wireless LAN, access point selection, wired path selection*

1. Introduction

Wireless LANs [1], which consist of access points (abbreviated by APs) and wireless stations, have become popular in recent years. Each AP connects to several other APs by wired links. Each station can use the wired network of APs as a backbone infrastructure by connecting to one of APs. That is, when a station sends a packet to another station (destination), it has to select an AP and a wired path (sequence of wired links) from the selected AP to the AP its destination connects to. Then, the station sends packets to its AP, and APs relay packets through the selected wired path. In the following, we denote an AP that a station s_p connects to by *s_p's connection AP* and a wired path that s_p communicates over by *s_p's communication path*. Routing in wireless LANs is the problem that each station selects its connection AP and communication path that achieve the highest communication throughput. Routing is a fundamental problem for many applications, and thus efficient routing protocols are required.

One of the difficulties in routing is that each station should avoid congested communications links among numerous stations. In general, when numerous stations connect to the same AP, the throughputs of all wireless links from the AP are degraded. Similarly, when numerous packets are sent through the same wired link, the throughput of the wired link is degraded. Thus, each station should select its connection AP and communication path appropriately in order to avoid the degradation caused by congestion.

The other difficulty is that each station knows only local information. More precisely, each station only knows throughputs of available wireless links and the communication path it currently uses. In other words, each station does not know throughputs of non-communication paths. In addition, each station does not know even available wired paths from non-connection AP. The station can aware a wired path from an AP only by connecting to the AP. Thus, each station has to select its connection AP and communication path that avoid congestions of communications with only local information. Moreover, the throughput of the communication between each station and its destination dynamically changes due to selections of other stations and external environmental changes such as faults of links. Therefore, highly adaptive routing protocols are required.

Many protocols for AP selection [6]–[8] and for wired path selection [9], [10] are proposed independently. We can realize a routing protocol for wireless LANs by naively combining the protocols. That is, each station selects its connection AP according to an existing protocol for AP selection and then the station selects its communication path according to an existing protocol for wired path selection. However, existing protocols for AP selection only consider the throughputs of wireless links. Therefore, it's possible that throughput of all wired path from a connection AP with high throughput of wireless links may be low.

It is well known that the biologically inspired approaches have high adaptability to dynamic environmental changes. Attractor selection is the biological model first introduced by Kashiwagi et al. [2]. It models how E. coli cells adapt to changes in available nutrient even if molecular machinery for signal transduction to the DNA is unavailable. Attractor selection behaves according to activity (goodness) of current state. If attractor selection evaluates the current state as good, it keeps the current state with high probability. On the other hand, if attractor selection evaluates the current state as bad, it randomly changes its state so as to find a new good state. By such a simple mechanism, attractor selection

Manuscript received February 6, 2009.
Manuscript revised June 11, 2009.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Osaka-shi, 565–0871 Japan.
[††]The author is with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.

DOI: 10.1587/transinf.E92.D.2016

can adaptively select good state.

Leibnitz et al. proposed several protocols based on the attractor selection for routing in overlay networks [3], [4] and in mobile ad hoc networks [5]. Leibnitz's method can select a good route adaptively to environments when each station knows the information (e.g. packet delays) of all available routes. However, Leibnitz's method cannot be applied to routing in wireless LANs since each station does not know the throughputs of wired paths except for its communication path.

In this paper, we propose a routing protocol that can be applied to wireless LANs based on the attractor selection. Our contributions are summarized as follows:

(i) We improve attractor selection to behave with only local information such as the throughput of the communication path and the throughput of available wireless links. Thus, our protocol can be applied to routing in wireless LANs.

(ii) We design the protocol with combination of multiple attractor selections. The technique enables us to divide a problem into several simpler problems. Our protocol consists of two attractor selections (for AP selection and for wired path selection) and achieves adaptive routing. To the best of our knowledge, our protocol is the first one designed with combination of multiple attractor selections.

The outline of our protocol is that, a station selects its connection AP and communication path according to its own attractor selection. If the station communicates with low throughput, attractor selection induces the station to change its connection AP or communication path randomly. If the station communicates with high throughput, attractor selection induces the station to continue to select its current connection AP and communication path with high probability. By this mechanism, the station can adaptively select its connection AP and communication path with high throughput.

The remaining of the paper is organized as follows. We describe the communication model in Sect. 2. Then, we explain the original attractor selection model and Leibnitz's method in Sect. 3. In Sect. 4, we propose a routing protocol for wireless LANs based on attractor selection. Then, we show the simulation results and discuss them in Sect. 5. Finally, we present concluding remarks in Sect. 6.

2. Preliminaries

2.1 System Models

A system consists of stations $S = \{s_1, s_2, \ldots, s_{|S|}\}$, APs $A = \{a_1, a_2, \ldots, a_{|A|}\}$ and a set L of bidirectional wired links, each of which connects to two distinct APs. APs a_i and a_j can communicate with each other directly only when a link $e_{i,j}$ (or a link $e_{j,i}$) is contained in L. APs a_i and a_j are called neighbors with each other if and only if $e_{i,j} \in L$. We define a wired path from an AP a_i to an AP a_j as a sequence of wired links $r = \langle e_{i_0,i_1}, e_{i_1,i_2}, \ldots, e_{i_{k-1},i_k} \rangle$ such that $i_0 = i$, $i_k = j$, and $e_{i_m,i_{m+1}} \in L$ $(0 \le m \le k-1)$. APs and stations are deployed on a two dimensional plane. Each AP a_i has communication radius c_i. A station s_p can connect to a_i by a wireless link when the distance from s_p to a_i is c_i or less. Let A_p be the set of APs that a station s_p can connect to. A station s_p can connect to at most one AP in A_p at the same time. We assume that stations cannot communicate with each other directly. This implies that, when a station sends a packet to another station, one or more APs have to relay the packet. To realize the communication from stations s_p to s_q, s_p has to connect to an AP a_i in A_p and select a wired path from a_i to s_q's connection AP.

In what follows, we define the throughput of communication from a station s_p to a station s_q. Let a_i and a_j be the connection APs of s_p and s_q, respectively. An AP a_i (resp. a_j) has a capacity TA_i (resp. TA_j) that represents the maximum wireless throughput of a_i (resp. a_j). Let $r = \langle e_{i_0,i_1}, e_{i_1,i_2}, \ldots, e_{i_{k-1},i_k} \rangle$ be a communication path from a_i to a_j. Every wired link $e_{g,h}$ has a capacity $TL_{g,h}$ that represents the maximum throughput of $e_{g,h}$.

The throughput of communication from the station s_p to the station s_q is defined as the minimum throughput on the wireless and wired links used for the communication. That is, the throughput $\theta_{p,q}$ of the communication from s_p to s_q is defined as follows:

$$\theta_{p,q} = \min\{\theta^a_{p,i}, \theta_r, \theta^a_{q,j}\},$$

where $\theta^a_{p,i}$ (resp. $\theta^a_{q,j}$) is the throughput of the wireless link between s_p and a_i (resp. s_q and a_j), and θ_r is the throughput of the wired path r.

Let NA_i be the number of stations that connect to AP a_i and $P_{p,i}$ be a packet error rate of the wireless link between the station s_p and a_i. Then, the throughput $\theta^a_{p,i}$ of the wireless link between s_p and a_i is defined as follows [6]:

$$\theta^a_{p,i} = \frac{TA_i \times (1 - P_{p,i})}{NA_i}.$$

Notice that a throughput of a wireless link between an AP and a station is degraded as the number of stations connecting to the AP increases.

The throughput θ_r of the wired path r is defined $\theta_r = \min\{\theta^e_{i_m,i_{m+1}} | 0 \le m \le k-1\}$, where $\theta^e_{i_m,i_{m+1}}$ is the throughput of the wired link $e_{i_m,i_{m+1}}$. Let $NL_{g,h}$ be the number of stations that communicate over wired paths including a wired link $e_{g,h}$. Then, the throughput $\theta^e_{g,h}$ of $e_{g,h}$ is defined as follows:

$$\theta^e_{g,h} = \frac{TL_{g,h}}{NL_{g,h}}.$$

Notice that a throughput of a wired link is degraded as the number of stations communicating over the wired paths including the wired link increases.

2.2 Routing Problem in Wireless LANs

In this paper, we consider the problem that each station selects its connection AP and communication path to maximize the throughput of each communication. We denote the destination station of a station s_p by $t(s_p)$.

In what follows, we define the available information for each station. We consider a situation that a station s_p connects to an AP a_i and communicates over a wired path r. In a large network, it is unrealistic to assume that each station knows all the information of the entire network. Thus, we assume that the station s_p knows only the following information:

- The throughput $\theta^a_{p,j}$ for each AP $a_j \in A_p$: The station s_p can compute the throughput because s_p can easily obtain NA_j, TA_j, and $P_{p,j}$ by the following way. Each AP a_j periodically sends NA_j and TA_j to all stations within the radius c_j. By periodical communications with AP a_j, s_p directly estimates the amount of lost packets. Then, s_p can calculate the packet error rate $P_{p,j}$.
- The throughput θ_r of the communication path r: The station s_p can observe the throughput θ_r by communicating over its communication path r.

In addition the station s_p receives the following information from its current connection AP a_i:

- The connection AP of the destination station.
- Wired paths from a_i to $t(s_p)$'s connection AP: Each AP can obtain this information by maintaining the routing table which contains several wired paths to other APs. We assume this routing table is constructed in advance by an existing protocol.
- The throughput $\theta^e_{i,j}$ for each neighbor a_j of the AP a_i (the throughput of the first step link in each wired path): Each AP periodically communicates with neighboring APs and records throughputs of these communications.

Notice that each station can connect to only one AP at the same time. Thus, each station does not know any information about wired paths from the other APs. Similarly, each station does not know the throughputs of wired paths except for its communication path.

In the following sections, we concentrate on the behavior of each station. That is, we propose a protocol that specifies which AP each station connects to and which wired path each station uses.

3. Attractor Selection

In this paper, we propose a biologically inspired routing protocol for wireless LANs. Our protocol is based on attractor selection, which is proposed by Kashiwagi et al. [2]. The original attractor selection models the behavior of E. coli cells that adapt to changes in the amount of available nutrient even if molecular machinery for signal transduction to the DNA is unavailable. Leibnitz et al. [3]–[5] extended the original 2-dimensional attractor selection to the M-dimensional one, and applied it to multi-path routing in overlay networks [3], [4] and in ad-hoc networks [5]. Our protocol is also based on Leibnitz's method. In this section, we explain principles of attractor selection and Leibnitz's method.

3.1 Principles of Attractor Selection

In the original biological model, attractor selection treats the concentrations m_i ($i \in \{1, 2\}$) of two kinds of mRNAs and activity α ($0 \leq \alpha \leq 1$). Activity α represents the goodness of current concentrations in the current environment. In E. coli cells, the concentrations of mRNAs are adaptively changed into the good ones, where the goodness is defined by activity. This behavior is represented as two differential equations:

$$\begin{aligned} \frac{dm_1}{dt} &= \frac{syn(\alpha)}{1+m_2^2} - deg(\alpha) \times m_1 + \eta_1, \\ \frac{dm_2}{dt} &= \frac{syn(\alpha)}{1+m_1^2} - deg(\alpha) \times m_2 + \eta_2. \end{aligned} \tag{1}$$

The term η_i is the white Gaussian noise. The functions $syn(\alpha)$ and $deg(\alpha)$ correspond to the rate coefficients of mRNAs synthesis and degradation, respectively. They are both functions of activity α and are defined as follows:

$$syn(\alpha) = \frac{6\alpha}{2+\alpha}, \qquad deg(\alpha) = \alpha. \tag{2}$$

The important character of expression (1) is that activity α changes the influence of a random term η_i. When the concentrations of mRNAs are not adapted to the external nutrient, activity α becomes close to 0. Then, the concentrations of mRNAs are randomly changed due to the random term η_i in expression (1). When the concentrations of mRNAs are adapted to the external nutrient, activity α becomes close to 1 and the concentrations of mRNA become stable.

3.2 Leibnitz's Method

Leibnitz et al. [3]–[5] extended the 2-dimensional attractor selection to the M-dimensional one, and proposed the method that can adaptively select one good state from M states according to the current environment. Since Leibnitz's method inherits adaptability from the original attractor selection, it can re-select a good state adaptively to the environmental changes. In Leibnitz's method, each m_i ($0 \leq i \leq M$) indicates that the i-th state is selected with probability $m_i / \sum_{k=1}^{M} m_k$. Activity α ($0 \leq \alpha \leq 1$) represents the goodness of the current value of each m_i ($0 \leq i \leq M$). The dynamic behavior of each m_i is represented as M differential equations:

$$\frac{dm_i}{dt} = \frac{syn(\alpha)}{1+m_{max}^2 - m_i^2} - deg(\alpha) \times m_i + \eta_i \qquad i = 1, \ldots, M, \tag{3}$$

where $m_{max} = \max\{m_i | 1 \leq i \leq M\}$. The functions $syn(\alpha)$ and $deg(\alpha)$ are defined as follows:

$$syn(\alpha) = \alpha \times (\beta\alpha^{\gamma} + \varphi^{*}), \qquad deg(\alpha) = \alpha, \tag{4}$$

where the parameters β and γ are the factors which control the influence of activity to changes of m_i. In [5], Leibnitz's method adjusts $\beta = 50$ and $\gamma = 3$, and we use the same parameters in this paper. The value φ^* is a special offset which we describe in detail later. For simplicity, we define $\varphi(\alpha) = syn(\alpha)/deg(\alpha)$. Then, the equilibrium solution of expression (3) is the following:

$$m_i = \begin{cases} \varphi(\alpha) & m_i = m_{max}\,(H\text{-value}) \\ \frac{1}{2}\left[\sqrt{4+\varphi(\alpha)^2} - \varphi(\alpha)\right] & m_i \neq m_{max}\,(L\text{-value}). \end{cases} \tag{5}$$

From expression (5), Leibnitz's method selects the c-th state such that $m_c = m_{max}$ with high probability and rarely selects the others. We call the c-th state such that $m_c = m_{max}$ as the inclined state. The value $\varphi(\alpha)$ needs to satisfy $\varphi(\alpha) \geq 1/\sqrt{2}$ in order that H-value becomes larger than or equal to L-value. Leibnitz's method adjusts $\varphi^* = 1/\sqrt{2}$, whereby H-value is equal to L-value when activity $\alpha = 0$.

The behavior of activity α is different in each method in [3], [4] and [5]. Thus, we explain only overview here. In all the methods, attractor selection calculates activity α to reflect the evaluation value of the inclined state compared to the best one. Therefore, attractor selection needs the evaluation values (e.g., packet delay) of all states. When the inclined state is not good for the current environment, activity α becomes closer to 0. Then, the dynamic behavior of m_i is strongly dominated by random term η_i, and the difference between H-value and L-value becomes smaller. Therefore, the system randomly changes the inclined state and selects all states with uniform probability. On the other hand, when the inclined state is good for the current environment, activity α becomes closer to 1. Then, the dynamic behavior of m_i is not influenced by random term η_i and the difference between H-value and L-value becomes larger. Therefore, the inclined state becomes stable and the system remains at the inclined state with higher probability. By this mechanism, Leibnitz's method can select a good state adaptively for the dynamic environment.

4. Our Protocol

In this paper, we propose a routing protocol in wireless LANs. Leibnitz's method [3]–[5] can find and keep a good state when each station knows the packet delays of all paths. However, we cannot directly apply Leibnitz's method to a routing problem in wireless LANs since each station knows only local information.

In our protocol, each station executes two attractor selections; one is for AP selection and the other is for wired path selection. The outline of our protocol is as follows:

- Each station s_p selects one AP from the set A_p based on its own attractor selection for AP selection. The evaluation value of an AP depends on whether the AP is its current connection AP or not.
  - If an AP a_i is s_p's current connection AP, the evaluation value of a_i is $\theta_{s_p,t(s_p)}$ that represents current throughput of the communication from s_p to $t(s_p)$.
  - If an AP a_i is not s_p's current connection AP, the evaluation value of a_i is $\theta^{a}_{p,i}$ that represents the current throughput of the wireless link between s_p and a_i. Note that the evaluation value does not depend on the throughput of wired paths from a_i. This is natural since a_i is not s_p's connection AP and s_p cannot know the throughputs.
- Each station s_p selects one wired path from the available wired paths (the wired paths from its current connection AP) based on its own attractor selection for wired path selection. Similar to attractor selection for AP selection, an evaluation value of a wired path differs whether the wired path is s_p's current communication path or not.
  - If a wired path r is s_p's current communication path, the evaluation value of r is its current throughput θ_r.
  - If a wired path r is not s_p's current communication path, the evaluation value of r is the throughput of the first step link in r. Note that the evaluation value does not depend on the throughput of r. This is natural since r is not s_p's communication path and s_p cannot know the throughput.

Both of the attractor selections in our protocol use only local information that each station knows. Thus, our protocol is suitable for distributed environments. In what follows, we explain the behaviors of the attractor selections.

4.1 Attractor Selection for AP Selection

In this section, we explain how each station s_p selects an AP from the set A_p. Let $a_1, a_2, \ldots, a_x$ be APs in the set A_p, where $x = |A_p|$.

4.1.1 Selection Manner

In our protocol, each station s_p selects an AP from A_p by attractor selection. Attractor selection returns real values $m_{1_a}, m_{2_a}, \ldots, m_{x_a}$. Each station s_p periodically selects an AP a_i with probability $m_{i_a}/\sum_{k=1}^{x} m_{k_a}$. Each value m_{i_a} is determined by other values m_{j_a} $(i \neq j)$ and activity α_a. Activity α_a, which is described hereinafter in detail, represents the goodness of an AP a_i that has the maximum value m_{i_a} among $m_{1_a}, m_{2_a}, \ldots, m_{x_a}$ (i.e., an AP that the station s_p most likely selects). We call such AP as inclined AP. The change of value m_{i_a} is defined as follows:

$$\frac{dm_{i_a}}{dt} = \frac{syn(\alpha_a)}{1 + (m_{max_a})^2 - (m_{i_a})^2} - deg(\alpha_a) \times m_{i_a} + \eta_{i_a} \qquad i = 1, \ldots, x, \tag{6}$$

where m_{max_a} is m value of the inclined AP and η_{i_a} is the white Gaussian noise. We define the functions $syn(\alpha_a)$ and

$deg(\alpha_a)$ as the same ones in expression (4). Let a_{max} be the inclined AP. As described in Sect. 3, the station s_p selects the AP a_{max} with higher probability than any other APs. By defining activity to reflect the goodness of the inclined AP, each station can connect to a good AP. Note that the station s_p does not necessarily select the AP a_{max} as the connection AP.

4.1.2 Behavior of Activity

Since a station s_p selects the AP a_{max} with high probability, we should reflect the goodness of a_{max} to activity α_a. When s_p connects to a_{max}, s_p knows the throughput of the communication from s_p to $t(s_p)$ via a_{max}. Thus, we reflect the current throughput of the communication to activity α_a. On the other hand, when s_p does not connect to a_{max}, s_p does not know the throughput of the communication from s_p to $t(s_p)$ via a_{max}. Thus, we reflect only the throughput of the wireless link between s_p and a_{max} to activity α_a.

While a station s_p connects to an AP a_{max}, activity α_a changes according to the following expression:

$$\frac{d\alpha_a}{dt} = \delta \times \left(\left(\frac{\theta_{p,t(s_p)}}{\max\{\theta_{p,i} | 1 \leq i \leq x\}} \right)^{\frac{u}{k}} - \alpha_a \right). \tag{7}$$

The parameters δ and u are the constants $\delta = 0.1$ and $u = 6$, and k is the dynamic value (described in detail later). Expression (7) shows that activity α_a converges to $\left(\frac{\theta_{p,t(s_p)}}{\max\{\theta_{p,i} | 1 \leq i \leq x\}} \right)^{\frac{u}{k}}$. Therefore, when the throughput $\theta_{p,t(s_p)}$ is much smaller than the maximum throughput of wireless links from s_p to APs, s_p regards a_{max} as a bad AP and activity α_a becomes close to 0. In contrast, when the throughput $\theta_{p,t(s_p)}$ is high enough compared to the maximum throughput of wireless links, s_p regards a_{max} as a good AP. Then, activity α_a becomes close to 1.

While a station s_p does not connect to an AP a_{max}, activity α_a changes according to the following expression:

$$\frac{d\alpha_a}{dt} = \delta \times \left(\left(\frac{\theta_{p,a_{max}}}{\max\{\theta_{p,i} | 1 \leq i \leq x\}} \right)^{\frac{u}{k}} - \alpha_a \right), \tag{8}$$

where $\theta_{p,a_{max}}$ is the throughput of the wireless link between s_p and a_{max}. Expression (8) shows activity α_a becomes high when the throughput of the wireless link between s_p and an AP a_{max} is high. Thus, the station s_p is likely to select an AP that has the wireless link with high throughput.

As described above, each station can find and keep a good AP. However, when the throughputs of all wired links in the networks are much lower than those of wireless links from s_p to APs, α_a never increases and thus each station frequently changes its connection AP. To avoid this situation, when a station frequently changes its connection AP, it suspects that all wired links have low throughput. Then, the station increments the variable k by one. Notice that, when k is large, activity α_a becomes less sensitive to the difference between the throughputs of wired links and wireless links. Therefore, by incrementing k, activity α_a is induced to increase. On the other hand, if a station continues to connect to the same AP over a certain period, it recognizes that the difference between the throughputs of wired links and wireless links is small. Then the station decrements the variable k by one while $k > 0$. In this paper, we adjust the variable $k = 2$ as the initial value in simulation settings.

4.2 Attractor Selection for Wired Path Selection

In this section, we explain how a station s_p selects its communication path after connecting to an AP. Let a_c and a_d be APs that stations s_p and s_q connect to. Let $r_1, r_2, \ldots, r_y$ be the wired paths between a_c and a_d. Let $e_1, e_2, \ldots, e_y$ be the first step links of wired paths $r_1, r_2, \ldots, r_y$, respectively.

4.2.1 Selection Manner

Similar to AP selection, attractor selection for wired path selection returns real values $m_{1_r}, m_{2_r}, \ldots, m_{y_r}$ and each station periodically selects a wired path r_i with probability $m_{i_r} / \sum_{k=1}^{y} m_{k_r}$. Each value m_{i_r} is determined by other values m_{j_r} $(i \neq j)$ and activity α_r. Activity α_r represents the goodness of a wired path r_i that has the maximum value m_{i_r} among $m_{1_r}, m_{2_r}, \ldots, m_{y_r}$ (i.e., a wired path that a station s_p most likely selects). We call such wired path as inclined path. The change of value m_{i_r} is defined as follows:

$$\frac{dm_{i_r}}{dt} = \frac{syn(\alpha_r)}{1 + (m_{max_r})^2 - (m_{i_r})^2} - deg(\alpha_r) \times m_{i_r} + \eta_{i_r} \qquad i = 1, \ldots, y, \tag{9}$$

where m_{max_r} is m value of inclined path and η_{i_r} is the white Gaussian noise. Let r_{max} is the inclined path. Expression (9) shows the same behavior as expression (6) and we omit further explanation.

4.2.2 Behavior of Activity

Similar to AP selection, our protocol reflects the goodness of the wired path r_{max} to activity α_r. While the wired path r_{max} is the communication path of the station s_p, activity α_r changes according to the following expression:

$$\frac{d\alpha_r}{dt} = \delta \times \left(\left(\frac{\theta_{r_{max}}}{\max\{\theta_{e_i} | 1 \leq i \leq y\}} \right)^{\frac{u}{k}} - \alpha_r \right), \tag{10}$$

where $\theta_{r_{max}}$ is the throughput of the wired path r_{max} and θ_{e_i} is the throughput of wired link e_i. The parameters δ, u and k are the same as those in expression (7). Expression (10) shows that activity α_r becomes high when the throughput $\theta_{r_{max}}$ is enough high compared to the throughputs of the first step links in other wired paths.

While the wired path r_{max} is not the communication path of a station s_p, activity α_r is changed according to the following expression:

$$\frac{d\alpha_r}{dt} = \delta \times \left(\left(\frac{\theta_{e_{max}}}{\max\{\theta_{e_i} | 1 \leq i \leq y\}} \right)^{\frac{u}{k}} - \alpha_r \right), \tag{11}$$

where $\theta_{e_{max}}$ is the throughput of the first step link in wired path r_{max}. Expression (11) shows that activity α_r becomes high when the throughput $\theta_{e_{max}}$ is high enough compared to those of the first step links in other wired paths.

5. Simulations

In this section, we show the performance of our protocol by simulations. Our simulation is done by computer experiment. We compare our protocol with a greedy-based protocol. The greedy-based protocol selects an AP based on Fukuda's protocol [6]. However, Fukuda's protocol solves only AP selection, and it does not consider the throughputs of wired paths. To consider the throughputs of wired paths, we change the evaluation of APs in Fukuda's protocol as follows. While a station s_p connects to an AP a_i, s_p evaluates a_i by the throughput of the communication from s_p to $t(s_p)$. When the station s_p changes its connection AP from an AP a_i to an AP a_j, it stores the throughput of the communication via a_i to use as the evaluation of a_i. The stored throughput is deleted after a certain period in order to update the information. While a station s_p does not connect to an AP a_i, s_p evaluates a_i by the stored throughput (if it exists) or the throughput of wireless link (if the stored throughput does not exist). Each station s_p connects to an AP whose evaluation is highest among A_{s_p}. After that, each station selects a wired path. Similar to AP selection, while a station s_p selects a wired path r, s_p evaluates r by its throughput. When the station s_p changes its communication path, it stores the throughput of r to use as the evaluation of r. A station evaluates wired paths by these stored throughput. If stored throughput of a wired path does not exist, a station evaluates the wired path by the throughput of the first step link in the path. When a station selects its communication path, it selects a wired path that has the highest throughput in its evaluations. If several wired paths have the highest throughput, one with the minimum number of hops is selected.

5.1 Simulation Settings

5.1.1 Routing Table

In this section, we explain the way to construct the routing table of APs in our simulations. When an AP a_i constructs the routing table to an AP a_j, a_i sends route request messages to all its neighbors. When some AP a_k receives the route request message from the AP a_i to the AP a_j, it forwards the route request message to all its neighbors. Although a_k may receive the same route request message multiple times, it forwards the message only for the first time. When the route request message is received by the destination AP a_j, a wired path from the AP a_i to the AP a_j is constructed. Notice that, since the route request message is forwarded via multiple paths, the AP a_i can construct multiple wired paths. The AP a_i stores at most four wired paths in the ascending order of the number of hops.

Table 1 The periods of storing throughputs in greedy-based protocols.

	periods of storing throughputs[rounds]		
	Greedy1	Greedy2	Greedy3
between stations	400~800	800~1600	1200~2400
wired path	80~160	160~320	240~480

5.1.2 Parameters

Our simulation environment consists of 15 areas, each of which is a 50 m × 50 m square field. In each area, there are 4 APs and 45 stations. APs are located at (12.5, 12.5), (37.5, 12.5), (12.5, 37.5), and (37.5, 37.5). Each AP has communication radius 40 m and capacity 50 Mbps. Each station is randomly positioned in an area. We define a packet error rate $P_{p,i}$ as $P_{p,i} = 0.8 \times (dist(s_p, a_i)/40)$, where $dist(s_p, a_i)$ is the distance between station s_p and AP a_i. Each station randomly selects its destination station and each source-destination pair is symmetric. The destination station of each station is decided at the beginning of each simulation and remains unchanged during the simulation. For both our protocol and the greedy-based protocol, each station is activated exactly once every round. Each station sends a packet to its destination station through its connection AP and communication path in every round. Each station re-selects its connection AP and communication path periodically. The interval between successive AP selections varies from 25 to 50 rounds, and the interval between successive wired path selections varies from 5 to 10 rounds. In the greedy-based protocol, each station stores the throughputs of the communications with its destination station and throughputs of wired paths during a certain period. In our simulation, we use three kinds of these settings (Greedy1, Greedy2, and Greedy3) to investigate the relation between the effectiveness of the protocol and the frequency of updating the information. Table 1 shows the period of each protocol. Each data of simulation is averaged over 100 simulation runs.

5.2 Network Environment

We present simulation results in two kinds of networks; one is a two-grouped network and one is a random network. In what follows, we explain each network environment.

5.2.1 Two-Grouped Network

The two-grouped network is constructed to evaluate the performance in the network where several bottle-neck links arise by environmental changes such as the failure of wired links. We verify that our protocol can appropriately select APs and wired paths to avoid bottle-neck links even in such a network. In the two-grouped network, APs are connected by wired links $e_{i,(i+2 \mod 60)}$, $e_{0,1}$, $e_{14,15}$, $e_{30,31}$, $e_{44,45}$ (see Fig. 1). That is, the network consists of two ring networks connected by four links. One ring consists of APs that have even suffix (even ring) and the other ring consists of APs that

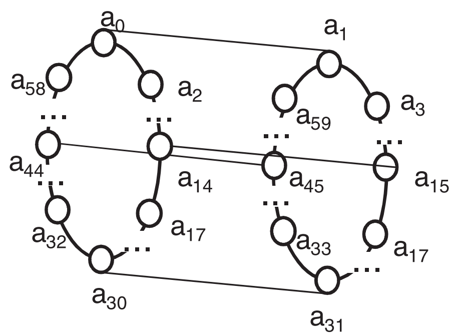

Fig. 1 The wired links of a two-grouped network.

have odd suffix (odd ring). APs are located randomly to satisfy that each area has two APs with odd suffix and two APs with even suffix. Our simulation executes 3000 rounds for each setting. We set the capacities of all wired links initially 500 Mbps. We degrade the capacities of all wired links in the odd ring to 100 Mbps after round 1000, and we restore the capacities of them to 500 Mbps after round 2000. While the capacities of all wired links in the odd ring are degraded, each station should connect to APs in the even ring since otherwise the throughput of communication becomes low. In the two-grouped network, the number of disjoint paths connecting two APs are at most three. Thus, each routing table has at most three paths (not four paths) that connect to an AP.

5.2.2 Random Network

The random network is constructed to evaluate the performance in non-biased networks. In the random networks, two distinct APs are connected by a wired link with constant probability so that the average degree of each AP becomes 3. In our simulations, we use only connected networks. Our simulation executes 3000 rounds for each setting. We set the capacities of all wired links initially 500 Mbps. We degrade the capacities of each wired link to 100 Mbps in probability 1/2 after round 1000, and we restore the capacities of wired links to 500 Mbps after round 2000.

5.3 Simulation Results

Table 2 shows the average throughputs of all communications in the two-grouped networks. Figure 2 shows the changes of the average throughput of each round in the two-grouped network.

In the two-grouped networks (Table 2), our protocol does not attain the worst performance in any condition. On the other hand, Greedy1 has the lowest throughput during rounds 1001 to 2000 although it attains the highest throughput during rounds 2001 to 3000. Similarly, Greedy3 has the lowest throughput during rounds 2001 to 3000 although it attains the highest throughput during rounds 1001 to 2000.

The results show that Greedy based protocols may attain good performance in special situation, but their performances are bad in other situations. On the other hand,

Table 2 Average throughputs in the two-grouped networks.

	Average throughput[Mbps]			
[round]	Our Protocol	Greedy1	Greedy2	Greedy3
1∼1000	4.01	3.78	3.78	3.78
1001∼2000	2.35	2.28	2.47	2.50
2001∼3000	3.86	3.98	3.40	3.17

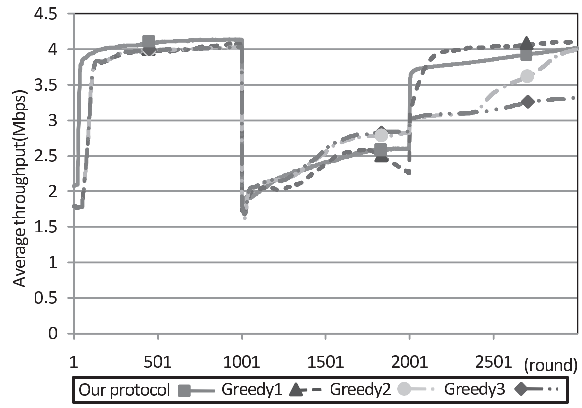

Fig. 2 Changes of average throughput in the two-grouped network.

our protocol can attain decent performance in any situation. These results show that our protocol has high adaptability to various environments.

In our protocol, each station selects its connection AP (communication path) by the goodness of current connection AP (communication path). Therefore, the throughputs of stations less depends on old information of the network environment. Thus, we can say our protocol has high adaptability against environmental changes. Figure 2 shows that our protocol gradually increases the throughput in rounds 1001 to 2000 and rapidly increases the average throughput after round 2000.

On the other hand, in the greedy-based protocols, each station selects its connection AP (communication path) by stored throughputs. Therefore, during rounds 1001 to 2000, each station can select good AP by greedy-based protocols once it detects the capacities of wired paths in the odd ring are degraded. However, in Greedy1, each station frequently discards stored throughputs and each station might connect to APs in the odd ring. Thus, Greedy1 attains the worst average throughput during rounds 1001 to 2000. Figure 2 shows that the greedy-based protocols also increase the throughput in rounds 1001 to 2000, however the average throughput of Greedy1 is readily degraded again. On the other hand, Greedy2 and Greedy3 store old information for a long time. Thus, most of stations connect to APs in even ring, despite that the wired links in the odd ring are restored after round 2000. Therefore, the average throughput of Greedy3 is the lowest after round 2000. Figure 2 shows that, after round 2000, Greedy1 rapidly increases the average throughput, however Greedy2 takes a long time until it increases the average throughput and Greedy3 does not increase the throughput.

To investigate stability of the proposed protocol, we

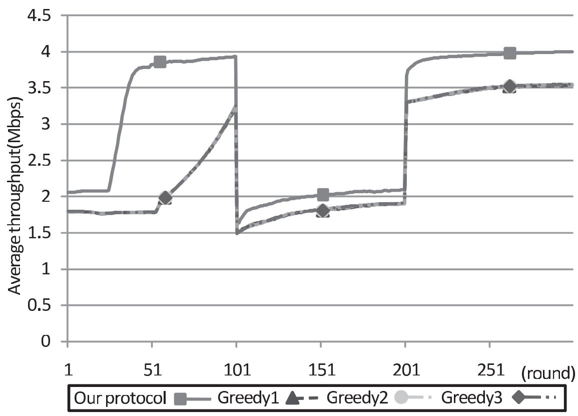

Fig. 3 Changes of average throughput in the two-grouped network. (300 rounds)

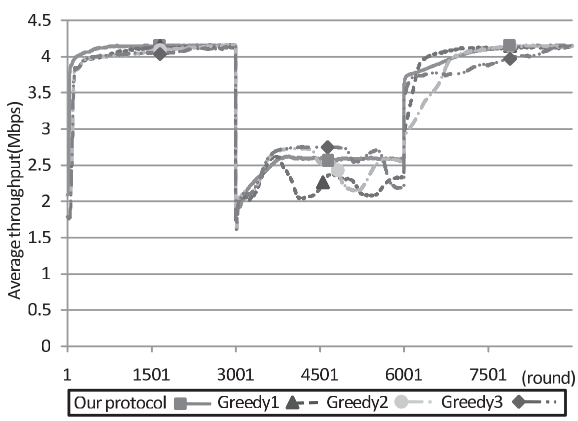

Fig. 4 Changes of average throughput in the two-grouped network. (9000 rounds)

also execute simulations with different frequencies of link capacity changes: one is simulation of 300 rounds with changing link capacities every 100 rounds, and the other is simulation of 9000 rounds with changing link capacities every 3000 rounds. Figures 3 and 4 show the results. The simulation results with the more frequent changes (Fig. 3) show that the average throughput of each protocol becomes lower. However, our protocol always keeps the highest throughput. We can conclude from the results that our protocol has rapid convergence and attain the highest adaptability to the frequent changes of link capacities. The simulation results with the less frequent changes (Fig. 4) shows that in rounds 3001 to 6000, all greedy-based protocols temporarily converge high throughputs, however, they degrade their throughputs after. The reason is that in 9000 rounds simulations, the period during which each station stores the information in greedy-based protocols are relatively short. Thus, each station discards the stored information several times and has to get the information again. On the other hand, once our protocol converges to high throughput, it keeps the high throughput until the environment changes.

To see the scalability of protocols, we also execute simulations with $|S| = 300$ and $|S| = 900$; this implies the

Table 3 Average throughputs in the two-grouped networks. ($|S| = 300$)

	Average throughput[Mbps]			
[round]	Our Protocol	Greedy1	Greedy2	Greedy3
1∼1000	7.80	7.42	7.44	7.45
1001∼2000	4.50	4.43	4.75	4.83
2001∼3000	7.54	7.76	6.59	6.11

Table 4 Average throughputs in the two-grouped networks. ($|S| = 900$)

	Average throughput[Mbps]			
[round]	Our Protocol	Greedy1	Greedy2	Greedy3
1∼1000	2.69	2.54	2.53	2.54
1001∼2000	1.61	1.54	1.67	1.69
2001∼3000	2.60	2.68	2.30	2.13

Table 5 Average throughputs in the random networks.

	Average throughput[Mbps]			
[round]	Our protocol	Greedy1	Greedy2	Greedy3
1∼1000	4.12	3.90	3.89	3.89
1001∼2000	3.65	3.61	3.65	3.64
2001∼3000	4.22	4.17	4.07	4.04

number of stations in each area changes to 20 and 60 respectively. The results presented in Tables 3 and 4 show that both the protocols vary throughputs almost in inverse proportion to the number of stations, which is the inevitable degradation in throughputs. Thus, we can say that both the protocols have scalability.

Table 5 shows that the difference of average throughputs is small in random network. In the random network, network diameter becomes small. Thus, there are many paths that have no bottle-neck link and each station can avoid bottle-neck paths regardless of protocol. However, our protocol always achieves the highest throughput in the random network. The results in the random network also show that our protocol has high adaptability to various environments.

6. Conclusion

In this paper, we have proposed a routing protocol for wireless LANs based on attractor selection. By applying the biologically inspired approach, our protocol has high adaptability to environmental changes. In our protocol, when a station communicates with low throughput, it randomly changes its connection AP or communication path. On the other hand, when a station communicates with high throughput, it continues to select its connection AP and communication path with high probability. As a result, our protocol can adaptively select its connection AP and communication path with only local information. We design the protocol with combination of multiple attractor selections and show the protocol performs effectively. We have shown by simulations that our protocol selects APs and wired paths so as to adapt to changes in the environment.

One of the future works is to investigate adaptability of the proposed method to mobile environment. We have already executed simple experiment. The result shows that there are little differences between non-mobile and mobile environment since movement of stations affects only the wireless throughput. Thus, we believe our protocol is also effective in mobile environment.

Acknowledgments

This work is supported in part by "Special Coordination Funds for Promoting Science and Technology: Yuragi Project" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, Grant-in-Aid for Scientific Research ((B) 17300020, (B) 19300017, (B) 20300012) of JSPS, Grant-in-Aid for Young Scientists ((B) 18700059) of JSPS, the Kayamori Foundation of Informational Science Advancement.

References

[1] B.P. Crow, I. Widjaja, J.G. Kim, and P.T. Sakai, "IEEE 802.11 wireless local area networks," IEEE Commun. Mag., vol.35, no.9, pp.116–126, Sept. 1997.
[2] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, "Adaptive response of a gene network to environmental changes by fitness-induced attractor selection," PLoS ONE, vol.1, no.1, p.e49, Dec. 2006.
[3] K. Leibnitz, N. Wakamiya, and M. Murata, "Resilient multi-path routing based on a biological attractor-selection scheme," Proc. 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT 2006), Jan. 2006.
[4] K. Leibnitz, N. Wakamiya, and M. Murata, "Biologically inspired adaptive multi-path routing in overlay networks," Proc. IFIP/IEEE International Workshop on Self-Managed Systems and Services (SelfMan 2005), May 2005.
[5] K. Leibnitz, N. Wakamiya, and M. Murata, "Self-adaptive ad-hoc/sensor network routing with attractor-selection," Proc. 49th annual IEEE Global Telecommunications Conference (GLOBECOM 2006), Nov. 2006.
[6] Y. Fukuda and Y. Oie, "Decentralized access point selection architecture for wireless LANs -deployability and robustness," Proc. 60th Vehicular Technology Conference (VTC2004-Fall), vol.2, pp.1103–1107, Sept. 2004.
[7] I. Papanikos and M. Logothetis, "A study on dynamic load balance for IEEE 802.11b wireless LAN," Proc. 8th International Conference on Advances in Communication and Control (COMCON 2001), June 2001.
[8] Y. Bejerano, S.J. Han, and Li (Erran) Li, "Fairness and load balancing in wireless LANS using association control," Proc. 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004), vol.2, pp.560–573, Sept. 2004.
[9] G. Apostolopoulos, S. Kamat R. Guerin, and S.K. Tripathi, "Quality of service based routing: A performance perspective," Proc. ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.17–28, Oct. 1998.
[10] S. Nelakuditi, Z.L. Zhang, and D.H.C. Du, "On selection of candidate paths for proportional routing," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol.44, no.1, pp.79–102, Jan. 2004.

Gen Nishikawa received the M.E. degree in computer science from Osaka University in 2007.

Tomoko Izumi received the B.E., M.E. and D.I degrees in computer science from Osaka University in 2003, 2005 and 2008. She had worked at Nagoya Institute of Technology as a postdoctoral researcher in 2008. She has been an Assistant Professor of College of Information Science and Engineering, Ritsumeikan University sicne 2009. Her research interests include distributed algorithms. She is a member of IEEE.

Fukuhito Ooshita received the M.E. and D.I. degrees in computer science from Osaka University in 2002 and 2006. Since 2003, he has been an Assistant Professor in the Graduate School of Information Science and Technology at Osaka University. His research interests include distributed algorithms and parallel algorithms. He is a member of ACM, IEEE, and IPSJ.

Hirotsugu Kakugawa received the B.E. degree in engineering in 1990 from Yamaguchi University, and the M.E. and D.E. degrees in information engineering in 1992, 1995 respectively from Hiroshima University. He is currently an associate professor of Osaka University. He is a member of the IEEE Computer Society and the Information Processing Society of Japan.

Toshimitsu Masuzawa received the B.E., M.E. and D.E. degrees in computer science from Osaka University in 1982, 1984 and 1987. He had worked at Osaka University during 1987–1994, and was an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) during 1994–2000. He is now a professor of Graduate School of Information Science and Technology, Osaka University. He was also a visiting associate professor of Department of Computer Science, Cornell University between 1993–1994. His research interests include distributed algorithms, parallel algorithms and graph theory. He is a member of ACM, IEEE, and IPSJ.