Resilient Multi-Path Routing Based on a Biological Attractor Selection Scheme

Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata

Graduate School of Information Science and Technology
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
[leibnitz,wakamiya,murata]@ist.osaka-u.ac.jp

Abstract. In this paper we propose a resilient scheme for multi-path routing using a biologically-inspired attractor selection method. The main advantage of this approach is that it is highly noise-tolerant and capable of operating in a very robust manner under changing environment conditions. We will apply an enhanced attractor selection model to multi-path routing in overlay networks and discuss some general properties of this approach based on numerical simulations. Furthermore, our proposal considers randomization in the path selection which reduces the selfishness and improves the overall network-wide performance.

1 Introduction

It is a well known fact that mechanisms found in biological systems are very robust and can handle changes in the environment very well. Therefore, many methods have been implemented in information science which mimic certain behavior found in nature. Some well known techniques like artificial neural networks, simulated annealing, or genetic algorithms are capable of performing well for certain problem types, especially in the presence of incomplete or fuzzy input data. In artificial neural networks, the concept of *attractors* is often used, which are equilibrium points or curves in the solution space to which the system converges depending on its initial condition. Attractors are a key issue in chaos theory and are often applied in mathematical models found in physics and bioinformatics.

Living organisms in nature continuously face a fluctuating environment and adaptation to these changing conditions is essential for the survival of the species. However, due to the high dimensionality of the habitat, each of the upcoming environmental changes rarely repeats itself during the lifetime of an individual organism. Therefore, the development of adaptation rules is not always feasible since learning and evolutionary processes require multiple occurrences of events to which the organisms adapt. Applying pattern-based learning techniques like in artificial neural networks is only possible, if input patterns and a desired target value exist. When no such input patterns exist, the adaptation to new situations is performed in a more self-organized manner. For example, cells can switch from one state to another depending on the availability of a nutrient [1]. These self-adaptive mechanisms are not necessarily optimal from the viewpoint of overall performance, but their main advantages lie in robustness and sustainability. This is a highly important feature for surviving in an unpredictable and fluctuating environment.

In this paper we extend the model of *adaptive response by attractor selection* (ARAS) which was introduced in [1] and apply it to the problem of multi-path routing. ARAS is originally a model for its host E. coli cells to adapt to changes in the availability of a nutrient for which no molecular machinery is available for signal transduction from the environment to the DNA. We will use this mechanism for switching between paths in a multi-path routing environment in communication networks. We consider an underlying IP layer with an overlay network in which an application specific routing is performed. This facilitates the implementation, as no modification to the existing IP layer is necessary. Each source may have several paths to the destination and splits its traffic depending on the current condition of the network over each path. However, one of the paths is chosen as primary path over which the majority of traffic will be routed, while the secondary paths are simply kept alive with a small proportion of the traffic. Attractor selection will be applied here to determine the primary path for a given traffic condition. When the environment, hence link qualities, changes such that the primary path is no longer appropriate, a new primary path is automatically selected. The advantage of our proposal is that there is no explicit routing rule for doing so, but everything is implicitly included in the differential equations describing the dynamics of the system. Furthermore, we use an inherent noise term to drive the system from one attractor to another, making the whole system also very stable to influences from noise.

The reason why we choose a dynamic system for self-adaptive routing instead of simple rule-based mechanisms is because our focus is on adaptiveness and stability of the system. Unlike most other routing papers like [2] which define a target function and perform an offline optimization of the OSPF weights using linear programming, we prefer a highly distributed sub-optimal solution, which is robust in the presence of fluctuations of environment conditions.

The remainder of this paper is organized as follows. In Section 2 we will briefly discuss the problem of multi-path routing in overlay networks and relevant work that is related to this topic. Then, in Section 3 we introduce the biological attractor selection model and extend the original model from $M = 2$ to a higher dimension. In Section 4 we illustrate how to use this proposed model for multi-path routing in overlay networks and we perform some simple simulations and discuss the results in Section 5. Finally, in Section 6 this paper is concluded with a short outlook on future work.

2 Related Work on Overlay Routing

Overlay networks have the appealing feature that their routing can be configured in an application-specific manner without modifying the underlying IP routing scheme. Before we discuss some related work, we would like to clarify the term of multi-path routing as we will use it in the following. The term multi-path routing has been used with different connotations. In all of them multiple paths are used from the source to destination over which traffic is transported. One interpretation of multi-path routing is to increase the resilience of the network, by simultaneously transmitting duplicates of the same packet over each path. This technique is often used in wireless ad-hoc networks [3] and it is sometimes referred to as *redundant multi-path routing*. Another way of using multiple paths is by distributing the traffic volume over these paths. Although by introducing this path diversity the routing is made more robust to failures of individual links, its main purpose is rather on performing load balancing [4]. We will use the latter notion of multi-path routing in our paper. An important issue in this type of multi-path routing which we will not address here is the topic of *packet reordering*, as some packets may overtake each other on different paths. The destination node must buffer the received packets and place them in the right order before delivering them to higher layers.

The issue of routing in overlay networks has been discussed, e.g. for *resilient overlay networks* (RON) [5], as an overlay network architecture which is able to improve the loss rate and throughput over conventional BGP routing due to its faster reaction to path outages. However, end-to-end route selection schemes as employed in overlay routing are of a highly selfish nature, as they greedily choose paths that offer the highest performance, regardless of the implications on the performance and stability of the whole network.

Several publications have investigated selfish routing using a game theoretical approach, e.g. [6, 7]. However, routing optimization is often performed with a global view of the network and its solution is computed by linear programming techniques. In our paper we wish to only consider the limited scope of information that a node can obtain from measurements of its links. In such a case, Seshadri and Katz [8] make suggestions to improve the overall stability of the system by imposing some restraints on the degree of selfishness of each flow. Randomization in path selection is one of such possibilities which we will also adopt in our approach. Another way to improve the overall system stability is to use a hysteresis threshold when updating the path decision.

User-optimal or selfish routing achieves a *Wardrop equilibrium* [9], which states that users do not have the incentives to unilaterally change their routes. Xie et al. [10] present a routing scheme which takes into account the user-optimal routing and network-optimal routing, where the former converges to the Wardrop equilibria and the latter to the minimum latency. In [11] an analytical model is constructed for multi-path routing which leads to an optimal number of links over which dynamic multi-path routing should be conducted. Su and de Veciana [11] propose a policy of routing the traffic to the set of least loaded links and show that this is especially suitable for high speed networks carrying bursty traffic flows.

An adaptive multi-path routing algorithm is proposed by Gojmerac et al. [12] that operates with simple data structures and is independent of the underlying network layer routing protocol. This is achieved by local signaling and load balancing resulting in the reduction of signaling overhead. Another measurement

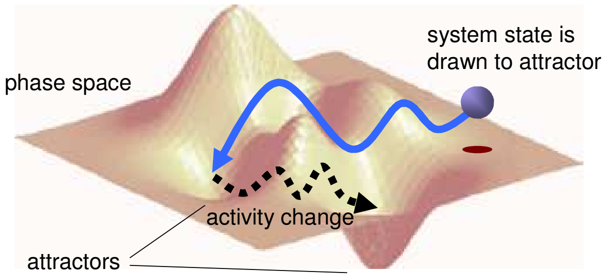


Fig. 1. General concept of attractor selection

based multi-path routing scheme is given by Güven et al. [13]. This method is similar to the work by Elwalid et al. [14], but does not require the explicit knowledge of the cost derivatives and due to stochastic approximation theory they use noisy estimates from measurements for estimating the cost derivatives. Other papers have dealt with improving the performance by sharing any unused other paths between different users. Approaches to MPLS [15] and WDM networks [16] have been proposed where the backoff capacity is shared, resulting in a better performance especially when supporting quality of service sensitive applications.

3 Biological Attractor-Selection Scheme

In this section we will give an outline of the principle of attractor-selection which is the key component in our method. The original model for adaptive response by attractor-selection is given by Kashiwagi et al. [1] and a first application to multi-path routing is performed in [17]. We will briefly summarize the basic method in an abstract problem formulation in this section, before introducing our extensions and discussing the proposed application to multi-path routing.

Basically, we can outline the attractor selection method as follows. Using a set of differential equations, we describe the dynamics of an M-dimensional system. Each differential equation has a stochastic influence from an inherent Gaussian noise term. Additionally, we introduce an *activity* α which changes the influences from the noise terms. For example, if $\alpha \to 1$ the system behaves rather deterministic and converges to attractor states defined by the structure of the differential equations, see Fig. 1. However, for $\alpha \to 0$ the noise term dominates the behavior of the system and essentially a random walk is performed. When the input values (*nutrients*) require the system to react to the modified environment conditions, activity α changes accordingly causing the system to search for a more suitable state (dotted line in Fig. 1). This can also involve that α causes the previously stable attractor to become unstable.

The random walk phase can be viewed as a random search for a new solution state and when it is found, α decreases and the system settles in this solution. This behavior is similar to the well known *simulated annealing* [18] optimization method, with the main difference that the temperature is not only cooled down, but also increased again when the environment changes.

3.1 Basic Biological Model

The biological model describes two mutually inhibitory operons where m_1 and m_2 are the concentrations of the mRNA that react to certain changes of nutrient in a cell. The basic functional behavior is described by a system of differential equations, see Eqns. (1).

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

The functions $syn(\alpha)$ and $deg(\alpha)$ are the rate coefficients of mRNA synthesis and degradation, respectively. They are both functions of α, which represents cell activity or vigor. The terms η_i are independent white noise inherent in gene expression.

The dynamic behavior of the activity α is given as:

$$\frac{d\alpha}{dt} = \frac{prod}{\prod_{i=1}^{M} \left[\left(\frac{nutr\_thread_i}{m_i + nutrient_i} \right)^{n_i} + 1 \right]} - cons\, \alpha, \tag{2}$$

where $prod$ and $cons$ are the rate coefficients of the production and consumption of α. The term $nutrient_i$ represents the external supplementation of nutrient i and $nutr\_thread_i$ and n_i are the threshold of the nutrient to the production of α and the sensitivity of nutrient i, respectively.

A crucial issue is the definition of the proper $syn(\alpha)$ and $deg(\alpha)$ functions. In our case, the ratio between $syn(\alpha)$ and $deg(\alpha)$ must be greater than 2 to have two different solutions of Eqn. (1) when there is a lack of one of the nutrients. When $\frac{syn(\alpha)}{deg(\alpha)} = 2$, there is only a single solution at $m_1 = m_2 = 1$. The functions $syn(\alpha)$ and $deg(\alpha)$ as given in [1] are shown in Eqn. (3).

$$syn(\alpha) = \frac{6\,\alpha}{2+\alpha} \qquad\qquad deg(\alpha) = \alpha \tag{3}$$

The system reacts to changes in the environment in such a way that when it lacks a certain nutrient i, it compensates for this loss by increasing the corresponding m_i value. This is done by modifying the influence of the random term η_i through α, see Fig. 2. When α is near 1, the equation system operates in a deterministic fashion. However, when α approaches 0, the system is dominated by the random terms η_i and it performs a random walk.

In Fig. 2 an example is given over 20000 time steps. We can recognize the following behavior. When both m_i values are equal, the activity is highest and $\alpha = 1$. As soon as there is a lack of the first nutrient ($2000 \leq t < 8000$), m_1

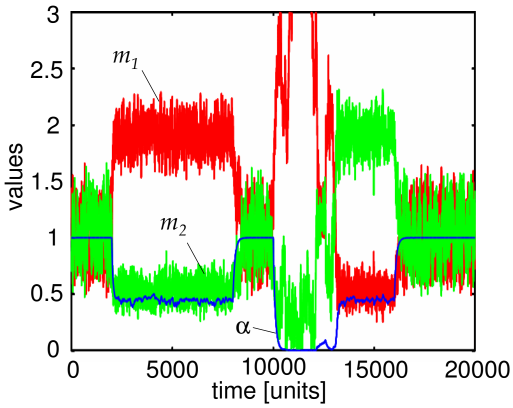


Fig. 2. Biological attractor selection model

compensates this by increasing its level. When both nutrient terms are fully available again ($8000 < t \leq 10000$), the activity α becomes 1 again. An interesting feature of this method can be observed between $10000 < t < 13000$. Here, the random walk causes the system to search for a new solution, however, it first follows a wrong "direction" causing α to become nearly 0 and the noise influence is highest. As soon as the system approaches the direction toward the correct solution again, α recovers and the system gets stable again. Such phases may always occur in the random search phase.

3.2 Multi-Dimensional Attractor Selection Model

In its original form, the attractor selection model only takes a dimension of $M = 2$ into account. Let us now consider a system of $M > 2$ equations as shown in Eqn. (4). The difference to Eqn. (1) is that we now have in the denominator the difference of the m_i value from its maximum $\hat{m} = \max_j m_j$. This does not fully have the direct mutual inhibitory effect anymore like in the original biological model, but makes it easier to extend.

$$\frac{dm_i}{dt} = \frac{syn(\alpha)}{1 + \hat{m}^2 - m_i^2} - deg(\alpha)\, m_i + \eta_i \qquad i = 1, \ldots, M \tag{4}$$

Furthermore, for the sake of simplicity we define in the following:

$$\varphi(\alpha) = \frac{syn(\alpha)}{deg(\alpha)}. \tag{5}$$

Equilibrium Points The equilibrium points have the condition

$$\frac{dm_i}{dt} = 0 \qquad \forall i = 1, \ldots, M$$

and can be easily computed from (4) when we assume without restriction of generality that m_i is maximal for an index $i = k$. Inserting this into Eqn. (4) we obtain M resulting vectors of the type

$$\mathbf{x}^{(k)} = \left[x_1^{(k)}, \ldots, x_M^{(k)}\right]^T \qquad\qquad k = 1, \ldots, M$$

with components

$$x_i^{(k)} = \begin{cases} \varphi(\alpha) & i = k \\ \frac{1}{2}\left[\sqrt{4+\varphi(\alpha)^2} - \varphi(\alpha)\right] & i \neq k \end{cases} \tag{6}$$

These results are all of the type

$$\mathbf{x}^{(k)} = [L, \ldots, L, H, L, \ldots, L]$$

with a single high value H at the k-th entry and all others are a low value L. Note that at

$$\varphi^* = \frac{1}{\sqrt{2}} \tag{7}$$

we have a special point, as the solutions $\mathbf{x}^{(k)}$ are only defined when $\varphi(\alpha) \geq \varphi^*$. For $\varphi(\alpha) = \varphi^*$ we obtain a single solution $\mathbf{x}$ with the same entries.

$$\mathbf{x} = [x_1, \ldots, x_M] \qquad \text{with} \quad x_i = \varphi(\alpha) \quad \forall i = 1, \ldots, M.$$

This structure of solution vectors is extremely useful to indicate that from all possible M paths, the k-th path is chosen as primary path or there is no specific primary path and the traffic is equally split among all paths.

Determination of the Activity Dynamics To fully specify the model, we need to define the basic dynamic behavior of the activity α and the functions $syn(\alpha)$ and $deg(\alpha)$. The eigenvalues of the Jacobian matrix at the solutions $\mathbf{x}^{(k)}$ always reveal negative values, leading to stable attractors [19].

Recalling the original biological model, we could identify three distinct stages during the convergence process: there was case (i) when all m_i were nearly equal due to a balanced condition at $\alpha = 1$. Then, there was case (ii) with one m_i taking a high value and the other m_j with $j \neq i$ a low value. In this case we had different attractor locations and the activity α was fixed at some level between 0 and 1. Finally, in case (iii) with activity $\alpha = 0$, we only had random influence.

In the following, we will slightly modify this general behavior. Our goal is to almost always perform a selection of a primary path out of the M possible paths. We will therefore definitely need case (ii) stated above. However, we merge cases (i) and (iii) to consider the scenario when all paths are nearly equal and we don't have a preference; we still choose one of them rather randomly as a primary path. Therefore, this modified method will always yield a primary path except for the time when a new solution is searched. Additionally, we shift the domain for α to the interval $[1, 2]$, since at $\alpha = 1$, we have the lowest absolute value of α and the highest influence from noise. On the other hand, all m_i are at the same value $\varphi(\alpha)$ which helps to recover from this state of equality among the paths and quickly drives one path to become the primary path.

Based on the above mentioned constraints, the quotient $\varphi(\alpha)$ should be a increasing function in $[1, 2]$ with $\varphi(1) = \varphi^*$. We use the following function given in (8).

$$syn(\alpha) = \alpha \left[(\alpha - 1)^2 + \varphi^* \right] \qquad\qquad deg(\alpha) = \alpha \tag{8}$$

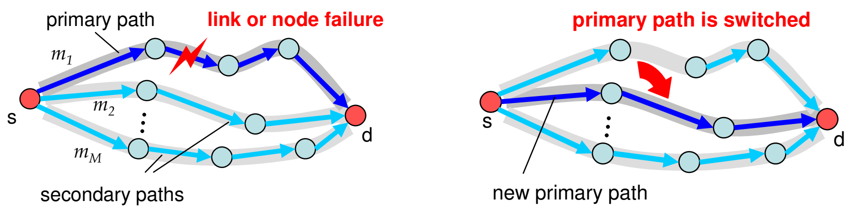
Let us now discuss the desired behavior of α. In order to specify its behavior, we must define what activity should indicate. In this paper, we consider the transmission delay on path i as performance metric l_i, so a "better" path is characterized by a smaller value of l_i. The output values m_i should reflect them by considering the minimum values of l_i. Hence, when an $\check{l} = l_k$ is the minimum of all input values, we wish that the system obtains m_k maximally. The dynamics of the activity behavior is shown in Eqn. (9). We introduce with Δ a hysteresis threshold in order to limit unnecessary oscillations between paths. The use of such a hysteresis was reported in [8] to reduce the selfishness and help improve the overall system performance.

$$\frac{d\alpha}{dt} = \delta \left(\left[\prod_{i=1}^{M} \left(\left(\frac{m_i}{\hat{m}} \frac{\check{l}}{l_i + \Delta} \right)^n + 1 \right) \right]^\beta - \alpha \right) \tag{9}$$

Like in the original model, the rate δ corresponds to the growth (*prod*) and decay (*cons*) rate of α, which we choose to be equal at $\delta = 0.01$. The parameter n given here, is an exponent which must be selected very large, e.g. $n = 100$ in order to "filter out" any unwanted intermediate values. Furthermore, we scale the output levels for H and L with the exponent β. A value of $\beta = 1.75$ has proven to be most effective. Within the product in (9) we could also add further input parameters for evaluating the current system condition in greater detail.

4 Application to Multi-Path Routing

The main problem that we focus on here is that for a certain source-destination pair, exactly one path is chosen as primary path based on the current environment condition. When the situation changes and the current primary path is no longer the best choice, the scheme adapts to selecting a different primary path which is better suited. The desired behavior is shown in Fig. 3. There are M paths from source s to destination d and one of these is the primary path over which the main traffic volume is transported. If a link or node fails on this path, the primary path is automatically switched to the best secondary path. The switching of paths should not only occur in such drastic conditions as



(a) Failure of primary path

(b) Switching to secondary path

Fig. 3. Desired behavior of routing method

link failures, but also of course when due to changed load conditions one of the secondary paths seems more appropriate as primary path.

The basic sequence of the routing algorithm consists of two steps: (i) route setup phase and (ii) the route maintenance phase. In the following sections we will discuss the operation of both of these phases.

4.1 Route Setup Phase

In the route setup phase we use a decentralized method similarly like in AODV routing. When a request for a new route to a destination arrives at the source node, it broadcasts *route request* (RREQ) packets to the overlay network. When a neighboring node receives an RREQ message and it has no route to the destination, it continues broadcasting the packet to its neighbors. However, if it receives an RREQ message that it has already processed, the request is discarded. In case the RREQ packet arrives at the destination node or another node which already has a route to the destination stored in its table, it replies with a *route reply* (RREP) packet to the source node requesting the route. As soon as the first RREP message arrives at the source it will have knowledge of a route to the destination node and will start using this route in its transmission. In such a way up to M routes are collected gradually and the route maintenance phase with the attractor selection algorithm will proceed with these M paths.

The route setup phase is initiated when the transmission request to an unknown node arrives at the source. After that the route maintenance phase is entered, in which the scheme will operate most of the time. However, in the case that paths are lost in the course of that phase and a minimum threshold of M_{min} is reached, route setup for additional paths is again invoked to add new paths.

4.2 Route Maintenance Phase

Once the first path from source to destination has been established, the route maintenance phase is performed. In this phase, the attractor selection model introduced in Section 3 is used to select the primary path for transmitting packets. This selection is done according to the metric values of each path. We assume that the transmission delay obtained from measurements of the round trip time

(RTT) of each packet can be captured by inline measurements to reduce any overhead from active delay measurements.

The main problem in overlay network routing is that the best path is often chosen in an entirely selfish manner and the overall system performance is neglected. This may lead to undesired instability and oscillation in the network load. Seshadri and Katz [8] have studied this issue and suggest three restraints on this greedy behavior to improve the overall system-wide performance: (*i*) randomization in the route selections, (*ii*) route changes performed with a hysteresis threshold, and (*iii*) increase of the time interval between route changes. They present three extensions of simple greedy routing where the route selection for each packet is performed with randomization: ARAND, GRAND, and SRAND. The basic operation of these three methods is sketched below. Further details can be found in [8].

ARAND: The path is randomly selected from the set of potential path with probabilities proportional to their metric.
GRAND: The path is randomly selected from the best K potential paths.
SRAND: A subset of K paths is chosen from the potential paths among which the path with the highest metric is selected.

We can integrate randomization of path selections easily in our model, by using path transmission probabilities p_i which are obtained as normalized values of m_i.

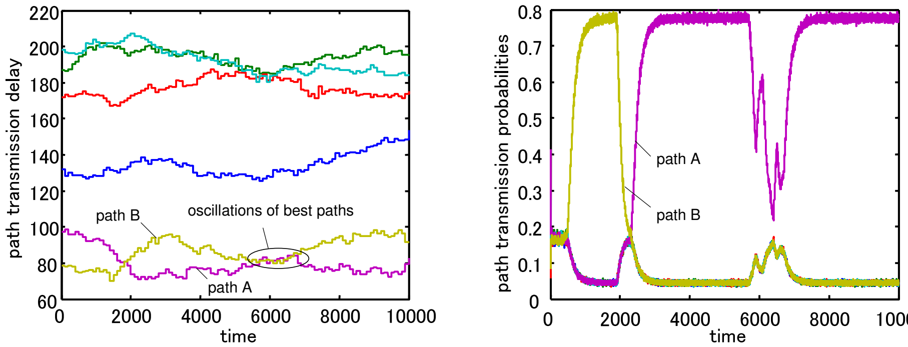
$$p_i = \frac{m_i}{\sum_{j=1}^{M} m_j} \qquad i = 1, \ldots, M \tag{10}$$

For this reason we will consider two variants of ARAS distinguishing between a probabilistic version and a deterministic version.

P-ARAS: The path is chosen with probabilities p_i.
D-ARAS: The path with the highest m_i level is selected.

An example of the input metric generated by a Wiener process is shown in Fig. 4(a) for each path and the resulting transmission probabilities for the P-ARAS method are given in Fig. 4(b). It can be seen that the transmission probabilities map well to the input metrics by choosing the path with minimum delay. At about time step 2000 the primary path is switched from path B to A. It can also be seen that although the path with best input metric oscillates between paths A and B around time step 6000, our method maintains path A as its primary path.

Using only a single metric value like in this case, makes the problem easy to tackle if we simply use a greedy approach, since there is an obvious mapping between input and output values. It should be emphasized, however, that our objective is not only to attempt to optimize the transmission delay of each individual user (as is done in the greedy case). By using randomization in the path selection we accept a slightly worse subjective performance in favor of an improved overall performance.



(a) Path delay as input metric (b) Transmission probabilities

Fig. 4. Input metric and transmission probabilities with P-ARAS

5 Numerical Results

In this section we will some discuss numerical results of our proposed method. The main performance metric we consider is the *average rate of path changes*. The *average transmission delay* would account only for the subjective performance, but we are more interested in observing the overall objective behavior. However, we will later also consider this metric.

Packets are generated in each slot with a certain probability p_{arr} which corresponds to a geometric time between arrival instants. For each packet arrival occurring at time t, the path over which it is transmitted is chosen by ARAS. If a path is selected that differs from the path used for the previous packet, we consider this a path change. Its total number is divided by the duration of each simulation run to obtain the path selection rate. A high value is, however, not necessarily an indicator for bad performance, since we assume that the paths have already been set up and there is no additional overhead for switching a path. It can be rather regarded as an indicator for the degree of path diversity. Clearly, a too high diversity results in a bad subjective performance since many "bad" paths are used and packet reordering may become necessary. On the other hand, a too small value indicates that the system operates rather deterministically. The whole problem narrows down to finding a good tradeoff between the user's subjective quality and the objective overall network performance.

Each simulation run has a duration of 10000 time steps and is repeated 1000 times. Since the confidence intervals are very small, we omit plotting them. We will focus our study on some of the parameter settings for the randomized version P-ARAS. The simulation scenario which we consider consists of a single source destination pair having $M = 6$ paths with metrics varying over time, see Fig. 5. The background traffic is modeled by initially uniformy distributed random path latencies and the evolution is performed by a Wiener process characterized by its standard deviation σ. Note that we will sometimes refer to this value simply

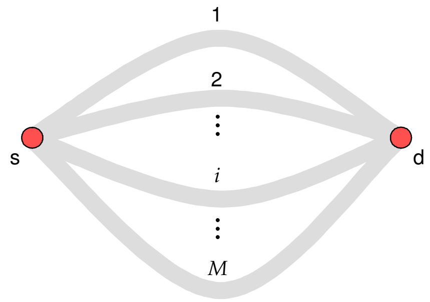


Fig. 5. Simple multi-path layout used in simulations

as variation of background traffic. We restrict the possible values of the path latency to be between a lower limit of 10 and an upper limit of 500.

5.1 Influence of Parameter Settings

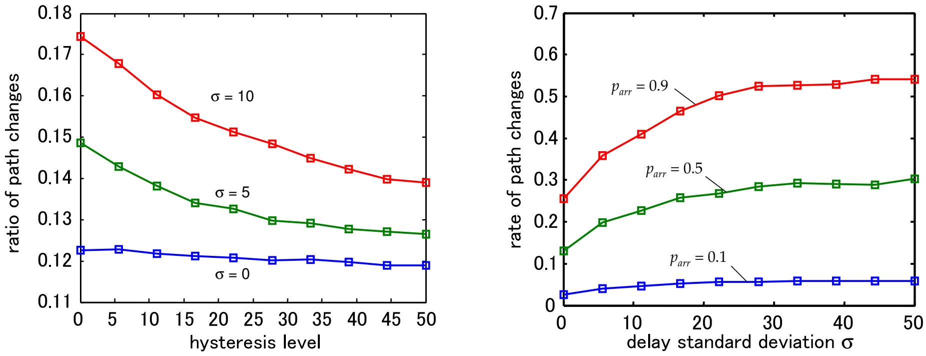
Let us consider at first the hysteresis threshold for switching paths. This parameter influences the reaction to sudden changes of the best path. However, when Δ is too large, the system becomes too slow in response to the metric changes and the performance degrades. The ratio of path changes is shown in Fig. 6(a) as a function of Δ.

The purpose of introducing the hysteresis threshold is to reduce the greediness by keeping the current primary path in spite of another one being slightly better. Using hysteresis shows a great advantage, especially when high oscillations among paths are observed . This is illustrated in Fig. 6(a) where the rate of path changes per packet is shown as a decreasing function over Δ. The slope of decrease becomes larger when σ is large.

In general, the hysteresis threshold should be selected depending on the variation of traffic, but the influence of an improper setting is not very crucial in the operation of our method. An algorithm for automatically selecting the hysteresis is proposed by the authors of [8] which could also be applied to our approach.

Next, we examine how the packet arrival rate influences the rate of path changes. Since we consider a discrete time system, we use a packet arrival probability p_{arr} in the simulation with geometrically distributed interarrival times. This corresponds to a Poisson arrival process with exponential interarrival time in the continuous time domain. We assume that the time steps are larger than the transmission time of the packets, leaving no direct interaction between the packets in this simulation scenario. Therefore, there is no influence of p_{arr} on the simulated average delay. The influence of p_{arr} on the rate of path changes is shown in Fig. 6(b).

The highest packet arrival probability causes also the highest path switching rate, as the arrival instants are more frequent and the sensitivity to traffic variations becomes larger. However, the curves flatten for large values of σ. This means that after the traffic variation reaches a certain level, it hardly influences



(a) Hysteresis level Δ (b) Packet arrival probability p_{arr}

Fig. 6. Influence of parameters on rate of path changes

the frequency of path switches. Although, the packet arrival probability does not influence the delay in our scenario, it does have an effect on the rate of path changes.

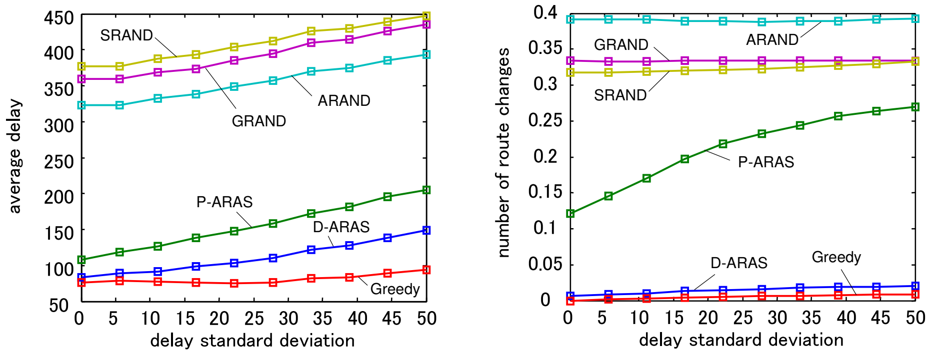
In this study we only consider a single flow from a source node to a destination node. When we extend our evaluation to a whole network with interacting flows in the future, we expect that the packet arrival rate will show some greater effect on the performance of our method.

5.2 Comparison of ARAS with Randomized Routing Methods

In the following we will compare the performance of D-ARAS and P-ARAS to the other methods introduced in Section 4. In general, there are two types of path selection methods, those with randomization and those without. While Greedy and D-ARAS are deterministic methods, all others use randomization for path selection. The subjective performance of the deterministic methods is naturally expected to be best, but they operate selfishly and thus are not efficient when considering the overall network performance.

Fig. 7(a) shows the average packet delay for each considered method in the presence of variation of the background traffic process. We use a packet arrival probability of $p_{arr} = 0.5$ and a hysteresis value of $\Delta = 5$. Greedy shows the expected best subjective performance with lowest delays. D-ARAS is only slightly higher, since it has a more delayed reaction than Greedy when choosing the paths. Of the randomized methods, P-ARAS is very efficient as it yields only slightly higher average delays than the deterministic algorithms. However, randomization clearly worsens the subjective performance experienced by the user's average end-to-end packet delay.

In Fig. 7(b) the rate of path changes is depicted. Obviously, the purpose of randomization is to balance the traffic among each path, so these methods yield a higher ratio. The Greedy method and D-ARAS have a very small ratio which



(a) Average total delay (b) Rate of path changes

Fig. 7. Comparison of ARAS with other methods

is caused by paths often staying best paths despite the presence of high variation of the others. Of the randomized methods again P-ARAS has the smallest path switching rate, whereas ARAND, GRAND, and SRAND stay nearly unaffected of the traffic variation. Clearly the highest path diversity is achieved by ARAND due to the proportional splitting of the traffic flow.

In general, we can show that P-ARAS is a good candidate for selecting paths, especially when we compare the results to the other randomized approaches. Its subjective performance reaches nearly that of the deterministic approaches while showing a high degree of path diversity.

6 Conclusion and Outlook

In this paper we presented an application of adaptive response by attractor selection (ARAS) to multi-path routing in overlay networks. ARAS is a biologically-inspired method and is robust to changes in the environment. The method converges to attractor solutions in the phase space and the selection of the appropriate attractor is driven by an activity term α. We have seen that by adequately defining the dynamic behavior of the activity α, we are able to map the input values to the selection of a primary path in an overlay network in a self-adaptive way.

Although the results suggested that the greedy approach appeared to show a good performance, the main drawback of using greedy path selection lies in the instability it introduces to the network. Whenever a new path appears more suitable, traffic flows are shifted and result in *route flapping*. For this reason, we implemented randomization of the path selections to reduce the greediness of each individual source-destination flow, while still achieving a good performance in terms of average packet delay. Furthermore, we investigated the influence of the key parameters of our model, such as the hysteresis threshold for switching paths under different levels of variation of the background traffic. Comparisons to other randomized methods showed the effectiveness of our approach. The main advantage of our proposal is that it operates without explicit rules and is simply implemented by numerical evaluation of the differential equations.

In the future, we wish to focus more on a network-wide view with large scale evaluations of the whole network. When evaluating the network in whole, we expect that our approach will be superior to the greedy method in performance. So far we considered only a single source-destination pair and the path selection was influenced only by the background traffic without any interaction from other flows operating with our method. As a main goal of further studies, we need to investigate and quantify the benefits of our proposed mechanism in the presence of interacting traffic. In such a way, the activity could be extended by some overall network performance metric resulting in a symbiotic selection of paths for each flow which is best for the whole network.

Acknowledgement

This research was supported by "The 21st Century COE Program: *New Information Technologies for Building a Networked Symbiosis Environment*" and a Grant-in-Aid for Scientific Research (A)(2) 16200003 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

1. Kashiwagi, A., Urabe, I., Kaneko, K., Yomo, T.: Adaptive response of a gene network to environmental changes by attractor selection. submitted for publication (2004)
2. Fortz, B., Thorup, M.: Internet traffic engineering by optimizing OSPF weights. In: IEEE INFOCOM, Tel-Aviv, Israel (2000) 519–528
3. Mueller, S., Tsang, R.P., Ghosal, D.: Multipath routing in mobile ad hoc networks: Issues and challenges. Lecture Notes in Computer Science **2965** (2004) 209–234
4. Andersen, D., Snoeren, A., Balakrishnan, H.: Best-path vs. multi-path overlay routing. In: Internet Measurement Conference (IMC), Miami Beach, FL (2003)
5. Andersen, D., Balakrishnan, H., Kaashoek, M., Morris, R.: Resilient overlay networks. In: 18th ACM Symposium on Operating Systems Principles (SOSP), Banff, Canada (2001)
6. Roughgarden, T., Tardos, E.: How bad is selfish routing? Journal of the ACM **49** (2002) 236–259
7. Qiu, L., Yang, Y., Zhang, Y., Shenker, S.: On selfish routing in internet-like environments. In: ACM SIGCOMM, Karlsruhe, Germany (2003)
8. Seshadri, M., Katz, R.: Dynamics of simultaneous overlay network routing. Technical Report UCB//CSD-03-1291, University of California, Berkeley, CA (2003)
9. Wardrop, J.: Some theoretical aspects of road traffic research, part II. In: Institution of Civil Engineers. Volume 1. (1952) 325–378
10. Xie, H., Qiu, L., Yang, Y.R., Zhang, Y.: On self adaptive routing in dynamic environments – an evaluation and design using a simple, probabilistic scheme. In: International Conference on Network Protocols (ICNP), Berlin, Germany (2004)

11. Su, X., de Veciana, G.: Dynamic multipath routing: asymptotic approximation and simulations. In: ACM SIGMETRICS, Cambridge, MA (2001) 25–36
12. Gojmerac, I., Ziegler, T., Ricciato, F., Reichl, P.: Adaptive multipath routing for dynamic traffic engineering. In: IEEE GLOBECOM, San Francisco, CA (2003)
13. Güven, T., Kommareddy, C., La, R.J., Shayman, M.A., Bhattacharjee, B.: Measurement based optimal multi-path routing. In: IEEE INFOCOM, Hong Kong (2004)
14. Elwalid, A., Jin, C., Low, S.H., Widjaja, I.: MATE: MPLS adaptive traffic engineering. In: IEEE INFOCOM, Anchorage, Alaska (2001) 1300–1309
15. Menth, M., Reifert, A., Milbrandt, J.: Self-Protecting Multipaths - A Simple and Resource-Efficient Protection Switching Mechanism for MPLS Networks. In: 3rd IFIP-TC6 Networking Conference (Networking), Athens, Greece (2004)
16. Gowda, S., Sivalingam, K.M.: Protection mechanisms for optical WDM networks based on wavelength converter multiplexing and backup path relocation techniques. In: IEEE INFOCOM, San Francisco, CA (2003)
17. Leibnitz, K., Wakamiya, N., Murata, M.: Biologically inspired adaptive multipath routing in overlay networks. In: IFIP/IEEE International Workshop on Self-Managed Systems & Services (SELFMAN 2005), Nice, France (2005)
18. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines. Wiley, New York (1989)
19. Murray, J.: Mathematical Biology, I: An introduction. 3 edn. Springer (2002)