

1

Biologically Inspired Networking

Kenji Leibnitz, Naoki Wakamiya and Masayuki Murata

Osaka University, Japan

1.1 Introduction

The development of computer networks has seen a paradigm shift from static, hierarchical network structures to highly distributed, autonomous systems without any form of centralized control. For networking nodes, the ability to self-adapt and self-organize in a changing environment has become a key issue. In conventional network structures, e.g., the Internet, there is usually a hierarchical order with centralized and static control. For example, hosts are aggregated to local area networks (LANs), which are connected via gateways to wide area networks (WANs) and network domains, etc., all using static connections and addressing. Recently, however, the trend leads more and more to networks that dynamically set up connections in an ad-hoc manner. Mobile ad-hoc networks (MANETs) are a prominent example, but also overlay structures such as peer-to-peer (P2P) networks require a scalable, robust and fully distributed operation with self-adaptive and self-organizing control mechanisms. The main control functions are no longer performed at intermediate nodes like routers, but shifted to the end-user nodes. Additionally, the location of these nodes may now be no longer static but can be mobile, imposing new challenges on the search for shared information in P2P networks or the location of a node in an ad hoc network.

For these types of new dynamic networks, the following three requirements for network control are considered mandatory:

- *Expandability (or scalability)*: facing a growing number of nodes and end users, as well as an increasing variety of devices attached to the network, the network must be able to continue its normal operation.
- *Mobility*: in addition to the end users' mobility, we should also consider the mobility and churn, i.e., the process of users entering and leaving the system, of intermediate

network nodes. This implies that deterministic packet forwarding cannot be expected, but must be performed in a probabilistic way.

- *Diversity*: the hardware and software of new types of network devices may generate entirely new traffic patterns imposing further challenges on the performance of the network. Additionally, networks must be able to cooperate with each other, both horizontally and vertically.

The only solution to meet the above characteristics seems to be that end hosts must be equipped with mechanisms permitting them to adapt to the current network status, for instance when finding peers or controlling congestion. For this reason, biologically inspired approaches seem promising since they are highly capable of self-adaptation, although they can be rather slow to adapt to environmental changes. Of course, the application of biologically inspired approaches in information technologies is not a new issue, but most of the previous attempts have been concentrated on optimization problems in network control. However, we can learn further important lessons from nature and the focus should lie on the scalability, adaptability, robustness and self-organization properties of biological systems. Especially, exploring the symbiotic nature of biological systems can result in valuable knowledge for computer networks.

The purpose of this chapter is to introduce general concepts from biological systems and to show their possible application in the field of computer networking. While nature in itself is of such high diversity and the topic of biologically inspired networking comprises many different aspects, only a small selection can be presented here. We will first discuss in Section 1.2 common principles found in biological networks and focus in particular on self-organization and the role of fluctuations. Then, we will show some analogies between biological processes found in the human body and computer networks. We will present some case studies of methods from swarm intelligence (Section 1.3), which describes the way individuals cooperatively interact with each other like swarms of bees or ants. Then, in Section 1.4 we will summarize some approaches which are based on evolutionary and adaptive systems, for instance featuring predator–prey-like behavior.

1.2 Principles of Biologically Inspired Networking

There are several key factors which can be observed in biological systems. Especially, features such as self-organization and robustness are of great importance when biological methods are applied to computer networks. However, there is also a trade-off to make when it comes to considering self-organized, distributed systems over those which are centrally controlled. Although scalability is improved, the approach towards fully distributed topologies comes at a cost of performance. Since there is no global view of the entire network, global optimization of network parameters is no longer feasible. Methods searching for local ad hoc solutions may yield only inferior results, so a trade-off must be found which balances scalability with controllability and performance.

Dressler [9] illustrates this trade-off as shown in Figure 1.1. The distinction is made between systems with centralized control and those that are fully self-organized. Distributed systems could be considered as intermediate, hybrid networks which allow the management of large numbers of nodes in a scalable way while preserving the benefits from a centralized control.

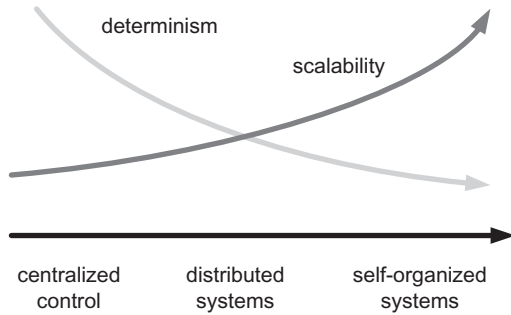


Figure 1.1 Trade-off between controllability and scalability in system control

Furthermore, while most computer networks are designed to optimally function under controlled conditions, biologically inspired systems rely on fluctuations and randomness. The benefits of the latter strategy can be seen when we consider the reaction to critical errors. Due to their probabilistic nature, they are less reliable on critical entities (e.g., servers, bottleneck links) with a single point of failure. Instead, several possible solutions may be considered as reaction to the failure, which although perhaps inferior in performance permits the operation at a sufficient quality. This leads to the behavior which can compensate critical errors, thus making the system more resilient and reliable, see Figure 1.2. However, biological systems usually do not instantly reach their operational performance level, but slowly approach this level by small fluctuations. This may require several (perhaps hundreds or thousands) of adaptation steps until reaching an acceptable level of performance.

1.2.1 Self-Organization

Perhaps the most important property of biological systems is their ability of self-organization, which describes systems consisting of autonomous individuals (cells, swarms of insects, etc.) grouping together into certain structures without any explicit rules. Self-organization is closely related to the emergence property of some biological systems, in which the outcome of the system depends on the collection of the individual behaviors

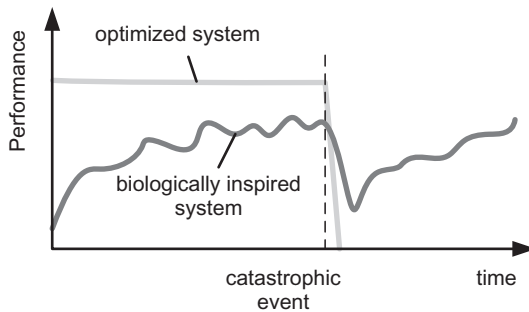


Figure 1.2 Biologically inspired systems may show inferior performance compared to optimized systems, but have a higher resilience towards critical errors

and their interaction and not so much on each individual task they perform. Bonabeau *et al.* [2] give a formal definition of self-organization as follows.

Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. The rules specifying the interactions among the system's constituent units are executed on the basis of purely logical information, without reference to the global pattern which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence.

Four basic principles can in general be observed in the self-organization of biological systems. *Positive feedback*, e.g., recruitment or reinforcement, permits the system to evolve, and promotes the creation of structure. Positive feedback acts as an amplifier for a desired outcome, whereas *negative feedback* regulates the influence from previous bad adaptations. Negative feedback also prevents the system from getting stuck in local solutions and may take the form of saturation, exhaustion or competition. Another important feature is that nature-inspired systems usually do not rely on any global control unit, but operate in an entirely distributed and autonomous manner. This means that each individual acquires its information, processes it and stores it locally. However, in order that a self-organized structure is generated, individuals need to exchange information with each other. This is done by either *direct or indirect interactions* among each other. Finally, a characteristic found in self-organized structures is that they often rely on *randomness and fluctuation* to enable the discovery of new solutions and to enhance the stability and resilience of the system.

1.2.2 Noise and Fluctuations

Biological systems are also characterized by the existence of an inherent stochastic component. In nature, nothing is perfect and there is no determinism. For example, if we consider cells of the same type, the values of the quantities describing them will vary from cell to cell and for a single cell, these values will also fluctuate over time [18]. When randomness is referred to in physics, we often consider the term of *Brownian motion*, which is the random motion of particles immersed in a fluid. The mathematical process used to describe it is called the *Wiener process*. Brownian motion is a continuous-time stochastic process and the position of the Brownian particle is described approximatively by the *Langevin equation*, see Equation (1.1).

$$\frac{dv}{dt} = -\gamma v + \eta \quad (1.1)$$

Here v is the velocity of the Brownian particle, γ is a friction coefficient of the underlying fluid, and η is a noise term. The Langevin equation can be solved by the Fokker–Planck equation to obtain the time evolution of the probability density function of the position and velocity of the particle.

1.2.2.1 Adaptive Response by Attractor Selection

Fluctuation can also be found as the driving force for information processing through the nonlinear behavior inherent to gene networks. These networks do not necessarily require

a preprogrammed connectivity for signal transduction from environmental inputs to DNA. For example, Kashiwagi *et al.* [19] introduce the concept of *adaptive response by attractor selection* (ARAS), which shows that a gene network composed of mutually inhibitory operons allows its host *Escherichia coli* cells to adapt to changes in the availability of a nutrient for which no molecular machinery is available for signal transduction. Since ARAS is driven by noise, it is noise-tolerant and can even be stimulated by noise. A cell activity or vigor leads to an alternative expression of the operon that produces the enzyme adaptively to the designated nutrient availability.

In ARAS, the concept of attractors is used to describe the multiple states of gene expression. An *attractor* is the region to which the orbit of a dynamical system recurrently returns regardless of the initial conditions [18]. Even if a state is perturbed by fluctuations, the system state will be drawn over time to an attractor. By formulating the desired outcome as a stochastic differential equation system and an appropriate mapping of the activity of the system, ARAS can be considered as an adaptive control mechanism with many possible areas of application. Leibnitz *et al.* [22] use ARAS to self-adaptively select the best paths in a multipath routing scenario in overlay networks. The method can be basically described as follows. The discovery of paths from source to destination is done in the *route setup* phase by broadcasting route request packets as in conventional routing methods, e.g., ad hoc on-demand distance vector (AODV) routing [25]. Then, in the *route maintenance* phase ARAS is used to determine the transmission probabilities for this set of paths. The addition or removal of paths is easily compensated.

Consider that there are M paths from source to destination. Then, for each packet sent from the source node, the path i is chosen with a probability or rate m_i which is determined by the stochastic differential equation system in (1.2).

$$\frac{dm_i}{dt} = f(m_1, \dots, m_M) \times \alpha + \eta_i \quad i = 1, \dots, M \quad (1.2)$$

Note that the basic structure of Equation (1.2) is similar to that in Equation (1.1). However, there is one important difference. Beside the function f which determines the attractor locations, the dynamics of each m_i depends also on the term α , which represents the *activity* of the system. Activity is directly influenced by the environment, which is in the case of the routing scenario expressed by the path metrics. So, when the activity approaches zero, the dynamic behavior of Equation (1.2) is dominated by the random noise term η_i , resulting in a random walk in the phase space. On the other hand, if α is large, the random component introduced by η_i becomes negligible and the system converges to an attractor solution in a deterministic way.

The transmission probabilities obtained by ARAS are illustrated in Figure 1.3 for an example scenario with $M = 6$ paths between a source node and destination node. The system automatically adapts to changes in the environment by searching for better solutions. In the case of the multipath scenario, this means that due to changes in the path metrics, the currently selected primary path is no longer appropriate, as can be seen at time steps 2000 and 6000 in Figure 1.3. The metric change causes activity to drop and the transmission probabilities approach each other until a better solution is found by random walk. In Figure 1.3, we can also recognize that the duration of the random walk phase varies for each search due to the memoryless search for a new solution. This causes a slight delay until the adaptation to the new solution is completed. The advantage, however, is that

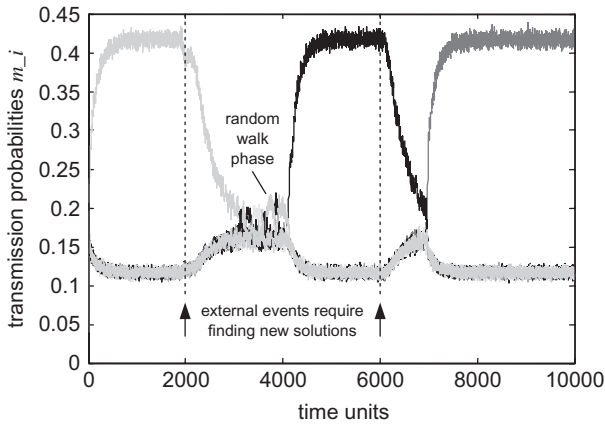


Figure 1.3 ARAS uses random walk to find new solutions when the environment conditions change

no explicit rule is necessary to trigger reactions to the environment since the behavior is implicitly encoded in the ARAS equations.

1.2.3 From Cell Biology to Computer Networks

In the field of cell and molecular biology, the behavior of cells is studied at a microscopic level. Cells are the smallest structural and functional units found in all living organisms and store their genetic information in the DNA. They take in nutrients and process these into energy (*metabolism*) and have the ability to reproduce. Basically, similar features are also found in autonomous systems in computer science [8].

1.2.3.1 Molecular Processes

Comparing organisms to computer networks, Dressler [8] states that there is a highly hierarchical structure found in both systems. Processes (e.g., movements) are organized by interactions of several organs, which are composed of tissues and these in turn are built by many different cells. One cell type consists of identical cells, which are associated and communicate with each other to fulfill a common function within the tissue.

Information exchange in a cellular environment is performed through signaling pathways from cell to cell, comparable to multihop transmission in communication networks. Cells communicate with each other by sending signal molecules which are bound by specific receptors and which are then forwarded to the cell nucleus causing a specific response, see Figure 1.4. The resulting cellular processes may mean that not only a single cell but several cells are activated, resulting in a coordinated reaction of the organ.

Several types of intercellular signaling exist depending on the distance over which the signal is transmitted. Over short range, there is *contact-dependent* signaling in which cells must have direct membrane-to-membrane contact and *paracrine* signaling where a local signal is diffused to react with multiple neighboring cells. On the other hand, when a signal needs to be transmitted over a long distance, signaling is performed either *synaptic* at the end of the axon or *endocrine*, by which hormones are distributed in the bloodstream

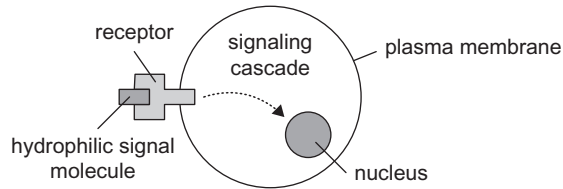


Figure 1.4 Local intercell communication is performed by binding signal molecules

of the body [1]. Mechanisms found in cell communication can be applied to computer networks in order to achieve a self-maintaining and self-healing approach to security in autonomous networking [20].

1.2.3.2 Artificial Immune Systems

Another similar approach for replicating biological behavior found in the body can be seen in the development of *artificial immune systems* (AIS). The main purpose of the biological immune system that has evolved in vertebrates is to protect the body from invading pathogens [4]. Its key function is to distinguish between *self* and *non-self* cells. Depending on the type and method of intrusion, the immune system uses different *immune response* mechanisms to destroy the invader or neutralize its effects. Biological immune systems consist of *innate immunity* and *adaptive immunity*. The innate immunity reacts to any pathogen that is recognized as being an intruder. If this fails, the body activates the adaptive immune system which has the ability to recognize and remember specific pathogens, and to mount stronger attacks each time the pathogen is encountered. Adaptive immunity uses two types of lymphocyte cells, B-cells and T-cells, which are created in the bone marrow and the thymus, respectively. Both types of cells are covered with antibodies which bind themselves to non-self antigens, see Figure 1.5.

The most important types of AIS are the *negative selection*, *danger signal*, *clonal selection* and *immune network models*. Negative selection is a mechanism to detect unknown antigens while not reacting to self-cells. Detectors are generated randomly and the thymus performs the negative selection by destroying those detectors that react against self-proteins. Those that do not bind self-proteins leave the thymus and act as memory cells circulating the body and protecting it from foreign antigens.

However, this self/non-self view does not fit experimental observations well, so the danger signal model [13] was introduced as an alternative approach. In the danger model, it is assumed that cells do not react directly to non-self elements, but the immune system

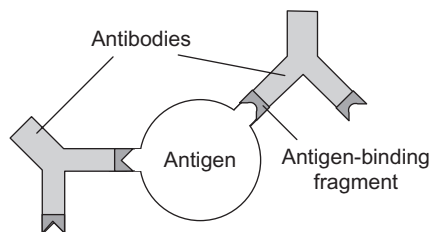


Figure 1.5 Immune response results in an intruding antigen being bound by different types of antibodies

responds to cells that raise an alarm to some form of danger. In this way, it can also be explained why the immune response does not react to harmless non-self cells, e.g., food, fetus. The danger signals are particularly triggered by *cell necrosis*, i.e., the unexpected death of cells, and the immune response is no longer a system-wide response, but must be seen in the context of the location of necrosis.

The immune system's ability to adapt its B-cells to new types of antigen is driven by clonal selection. Depending on the match strength between a B-cell and an antigen, B-cells are cloned and mutated with rates inversely proportional to the match strength (*affinity maturation*). This permits a more focused response to further attacks and the adaptation to changing non-self cells.

A network of interconnected B-cells for antigen recognition is the basis of the artificial immune network model. By stimulating and suppressing each other, a stable network between antibodies is formed which is capable of storing its state as immunological memory, comparable to a Hopfield neural network [17].

Artificial immune systems have been applied to intrusion detection and network security [12]. The term *computer virus* already indicates the analogy that can be found in computer networks and biological processes. Therefore, AIS are often used as mechanisms to detect virus infection of individual hosts by identifying abnormal behavior in terms of system calls (*host-based intrusion detection*). In the same way, TCP network traffic can be monitored for irregularities from the normal behavior (network-based intrusion detection). In [26] an AIS is constructed to detect misbehavior in a mobile ad hoc network, for example caused by faulty hardware or intentional disruption of the network. The model presented there uses a virtual thymus which causes the AIS to become tolerant to previously unseen normal behavior of the nodes, but automatically learns and detects new misbehavior. Further applications of AIS to computing and networking as well as to other fields can be found in [28].

In summary, artificial immune systems can provide a flexible and efficient method to monitor the integrity of a single host or a network and to react to potential dangers. Since the detector cells are small and lightweight, they can be efficiently used in a distributed environment, thus showing good scalability. This distributed operation also results in a very robust system, where the failure of a single detector cell does not result in the failure of the complete detection mechanism. Another useful characteristic is the adaptability found in AIS. The information obtained from distinguishing between self, harmless non-self, and harmful non-self remains in memory permitting a quick reaction to new unknown threats.

1.2.3.3 Bio-Networking Architectures

The application of principles from biology to computer network architectures is reflected in the bio-networking architecture [32]. It can be considered as both a paradigm and a middleware for constructing scalable, adaptive, and survivable applications [27]. The concept of emergence is taken into account by implementing network applications as a group of distributed and autonomous objects called *cyber-entities* (CE). The CE agents act autonomously based on local information and local interactions. Natural selection and evolution is performed and the CE must housekeep their energy in order to avoid starvation.

The bio-net platforms provide the execution environment for the CE, usually a Java virtual machine. Resources on a bio-net platform must be purchased by the CE with energy.

Furthermore, the bio-net is responsible for system services and information services to the CE as well as their scheduling. The self-managing features of the bio-networking architecture are also briefly discussed in Chapter 4.

BIONETS is a similar framework that was proposed for pervasive communication environments [3]. Its main differences from the architecture in [32] are that it is not limited to agent-based services and that the network connections and services are considered as evolutionary processes. BIONETS focuses on scalability, heterogeneity and complexity challenges found in mobile networks. On the lowest plane of a hierarchical structure, tiny nodes (*T-nodes*) gather data from the environment and pass them on to the user nodes (*U-nodes*) on the next plane. U-nodes form islands of connected devices and may mutually exchange information if they are within communication range. Services reside on the next higher plane and may also interact to reflect the social networks of the users.

1.3 Swarm Intelligence

In biology, the intelligent and well-organized behavior of a group of social insects can often be observed. For example, ants or bees solve complex tasks like building nests or searching for food by distributing simple tasks among each other. In such emergent systems, it is not so much the individual work which determines the outcome, but rather the collection of all single activities. This behavior is generally referred to as *swarm intelligence* [2]. The collaboration of insect societies is based on the principle of *division of labor* where specialized workers perform specialized tasks in parallel, leading to a better efficiency. Division of labor can be seen for all insect types in the distinction of worker and reproductive castes. In some cases, several different types of workers coexist which have distinctive tasks as well, e.g., protection of the nest or nurturing the brood.

Each individual insect makes its own decisions, solely based on locally available information. In other words, the insect is not aware of the global situation but only gains its input by interaction with other members of its species. This interaction may be either by direct contact (visual or chemical) or indirect interaction. In the latter case, one individual influences the environment and another reacts to this change at a later time. Ants, for example, interact with each other by laying a trail of pheromones which influences other ants to follow this trail, thus, reinforcing it. Such indirect interaction through the environment is called *stigmergy*.

1.3.1 Ant Colony Optimization

The *ant colony optimization* (ACO) meta-heuristic was introduced by Dorigo [7] as a method for probabilistic optimization in finding the shortest path in a graph. It is an extension of the *ant system* (AS) method based on the foraging behavior of ants that try to find their way between a food source and their nest. ACO can produce nearly optimal solutions to the *traveling salesman problem* (TSP) and its strength lies in the ability to dynamically adapt to changes in the graph topology. For this reason it is well suited for the application to routing in computer networks.

Basically, the method can be described as follows. Agents (artificial ants) randomly wander around searching for food and return to the nest when they have found a source, see Figure 1.6. They lay trails of pheromones which are also followed by other ants should they encounter them. As pheromones evaporate over time, unsuccessful paths are

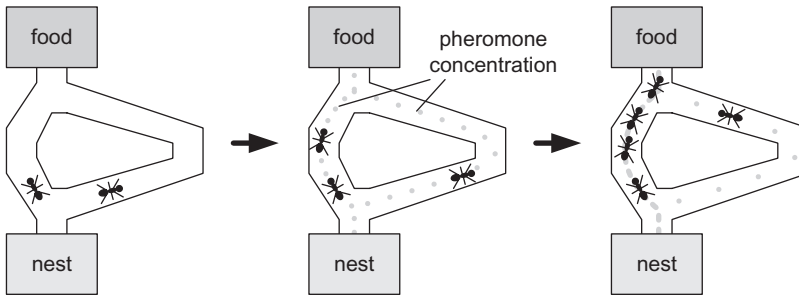


Figure 1.6 In ACO new paths are taken by ants randomly. As the pheromones evaporate over time, the probability for choosing unsuitable paths decreases

soon discarded which constitutes a negative feedback. On the other hand, there is positive feedback for successful paths leading to the food source as they are reinforced by other ants following the trail and increasing the pheromone density.

1.3.1.1 AntNet

AntNet by Di Caro and Dorigo [5] is an application of ACO to routing in packet-switched networks. The operation of AntNet is performed by two different types of ants. At regular intervals, each router in the network sends *forward ants* to probe the network for a minimal cost route towards randomly selected destinations. Whenever forward ants encounter a router on their way, they randomly choose the next hop depending on the probabilities in the routing table of the router. In the case that the destination is unknown at a router, the next hop is selected uniformly among all possible candidates. Once the forward ants reach their destination, they transform into *backward ants* following exactly the same path in the reverse direction to the source. During their return trip, they update the entries of the routing tables at each intermediate router according to the goodness of the path which they encountered. As soon as a backward ant reaches its originating node, it dies and is removed from the system.

1.3.1.2 AntHocNet

Due to the capabilities of operating in a changing environment, ACO also provides an effective principle for routing in mobile ad hoc networks. *AntHocNet* [6] is built on ACO routing and uses a hybrid approach to multipath routing. While AntNet uses a proactive scheme by periodically generating ants for all possible destinations, AntHocNet consists of both reactive and proactive components. Paths are only set up to destinations when they are needed by reactive forward ants that are launched by the source in order to find multiple paths to the destination. Backward ants return to the source to set up the paths. Data is routed stochastically over different paths stored in the pheromone tables. These tables are constantly updated by proactive forward ants. The algorithm reacts to link failures with either a local route repair or by warning preceding nodes on the paths. The authors show that AntHocNet can outperform AODV, which is a well-known routing protocol for ad hoc networks. Due to its emergent operation, AntHocNet also shows much better scalability than AODV.

AntNet and AntHocNet provide flexible and resilient methods for routing in packet-switched networks. The key point in both approaches is the use of probabilities for determining the next hop. These probabilities are constantly updated with the information from probing forward ants which register any changes in the network topology and thus make ACO-based routing very adaptive. However, the probabilistic search for the destination node may require a long time and a large number of forward ants if the network topology is large.

1.3.2 Synchronization with Pulse-Coupled Oscillators

Self-organized and fully distributed synchronization can also be found in nature where it is attained only by mutual interactions among individuals without the existence of a controlling unit. For example, groups of fireflies in Southeast Asia flash in synchrony. Each individual firefly flashes independently of each other at its own interval defined by an internal timer. However, when a firefly meets a group and observes flashes from other fireflies, it reacts to the perceived stimuli and adjusts its internal timer so as to flash at the same rate as the others.

Mutual synchronization of firefly flashing is modeled as a set of *pulse-coupled oscillators* [23]. Each oscillator has a state between zero and one, that monotonically increases with the phase of a timer, see Figure 1.7. The phase cyclically shifts from zero to one as time passes and when the state reaches one, the oscillator fires a pulse, after which it resets the phase back to zero. This happens at time t^+ immediately after the firing instant at time t . This notation is required, since oscillators react after they register the flashing of coupled oscillators, so a slight processing delay is necessary. The fired pulse stimulates other oscillators which are coupled to the firing oscillator and they react by updating their own state. After a series of such mutual interactions, the system achieves synchronization.

Wakamiya and Murata [31] apply a model of pulse-coupled oscillators to data gathering in wireless sensor networks. The purpose of using pulse-coupled oscillators is that each sensor node independently determines the cycle and timing at which it emits a message. For efficient data gathering, it has the effect that sensor information is propagated in concentric circles from the edge of the network to the base station to which all sensed data is transmitted. Thus, sensor nodes at a similar distance from the base station must transmit their information simultaneously, slightly before their inner neighboring nodes are about to transmit. The information and states of a sensor node i consist of its own

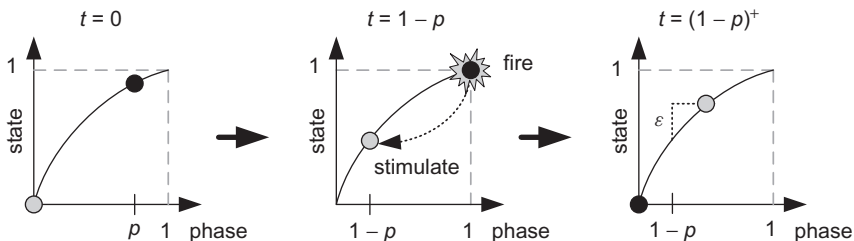


Figure 1.7 Pulse-coupled oscillator model with two oscillators. When one of them fires at time $t = 1 - p$, the other is stimulated causing a change in state

sensor data and that received from its neighbors, a phase φ_i of its timer, the state x_i defined by a monotonically increasing function $x_i = f(\varphi_i)$, a level l_i which corresponds to the number of hops from the base station, and an offset value δ .

Initially, none of the sensor nodes knows its distance to the base station. They broadcast sensor data at their own timing and frequency when the regulated state $x'_i = f(\varphi_i + \delta)$ reaches one. When a sensor node receives a radio signal with a level smaller than itself, it is stimulated and raises its state as shown in Equation (1.3).

$$x_i(t^+) = B(x_i(t) + \varepsilon) \quad \text{where} \quad B(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \\ x & \text{otherwise} \end{cases} \quad (1.3)$$

At the desired frequency of data gathering, the base station periodically broadcasts beacon signals within its RF transmission range. Sensor nodes which receive this signal recognize that they are within the innermost of the concentric circles around the base station and set their level to 1. In addition, by repeatedly being stimulated by the beacon signals, they become synchronized with these. A synchronized sensor node begins to transmit messages including sensor data and their level at the same frequency as the beacon signals, but at a time instant earlier than the beacon given by the regulated state x'_i . Sensor nodes receiving signals from a level-one node recognize themselves as being two hops away from the base station. In general, each node sets its own level to one level above the minimum from all received messages. Their timers get synchronized with the message transmissions of level-one nodes and they begin their own transmissions at 2δ earlier than the beacon signals. Their signals further stimulate outer nodes. Consequently, all sensor nodes identify their correct levels and behave in synchrony so that the waveform data gathering is accomplished.

Each sensor node can save battery power by turning on the transceiver only from the time it receives a message from its outer neighbors until it completes its own transmission to an inner neighbor. When a sensor node is moved to another location, it is stimulated by its new neighbors and joins the appropriate level of concentric circles. When a sensor node notices the disappearance of an inner node, it rejoins the network by synchronizing with another sensor node in its vicinity. The advantages of using such a synchronization scheme over other conventional methods are manifold. In this scheme, global synchronization is established and maintained only through mutual interactions among neighboring nodes. Each sensor node only transmits its sensor data at its own timing and the signals for data gathering are used for synchronization. Therefore, there is no need for any additional communication that may consume battery power and no time stamp is required for the messages. Finally, no additional routing protocol is needed which would also require additional signaling for path establishment and maintenance. However, it should be noted that this scheme also suffers from the same problems usually found in biological systems: global synchronization and reaction to changes in the environment due to failed sensors is never attained instantly, but only after a short delay.

1.3.3 Reaction–Diffusion

Pattern formation on the coat of mammals (e.g., spots of a leopard) and fish (stripes of an emperor angelfish) can be explained by reaction–diffusion (RD) equations. According

to Turing [30], these patterns can arise as a result of instabilities in the diffusion of morphogenetic chemicals in the animals' skins during the embryonic stage of development.

The two parts involved in this process are the long-ranged, slowly propagating diffusion effect of a chemical substance over space and the local reaction to it. Patterns are formed by chemical kinetics among two *morphogens*, the *activator* A and *inhibitor* B . Their concentrations, $A(\mathbf{z}, t)$ and $B(\mathbf{z}, t)$, at a specific location \mathbf{z} on the surface determine the state (or color) at time t . The mechanism is expressed by a pair of second order partial differential equations, where the concentration of one substance is defined by the reaction to the concentration of substances at the location and the respective diffusion components, D_A and D_B .

$$\begin{aligned}\frac{\partial A}{\partial t} &= F(A, B) + D_A \nabla^2 A \\ \frac{\partial B}{\partial t} &= G(A, B) + D_B \nabla^2 B\end{aligned}$$

The constants D_A and D_B with $D_B > D_A$ describe the speed of diffusion, the functions F and G are the nonlinear reaction kinetics, and the Laplacian $\nabla^2 A$ indicates the difference in concentration of morphogen A at a certain location with respect to surrounding concentrations. Depending on the sign of $\nabla^2 A$, morphogen A will either diffuse toward that nearby location or away from it.

Beginning from a homogeneous and even distribution of chemical substances over a surface and by regulating initial random influences and parameters in the RD equations, a variety of patterns (stripes, spots, rings) can be formed by evaluating the equations at each point independently. Figure 1.8 shows some sample patterns generated from the reaction–diffusion equations with different parameter settings.

Reaction–diffusion has been applied to several different areas in computer science and communication networks. In computer graphics, reaction–diffusion is used to generate realistic textures on surfaces. Other cases include using RD for solving optimization problems. Yoshida *et al.* [33] propose using reaction–diffusion as a cooperative control mechanism in a surveillance system of cameras. Each camera adjusts its observation area in a decentralized way based on reaction–diffusion equations to reduce blind spots. The

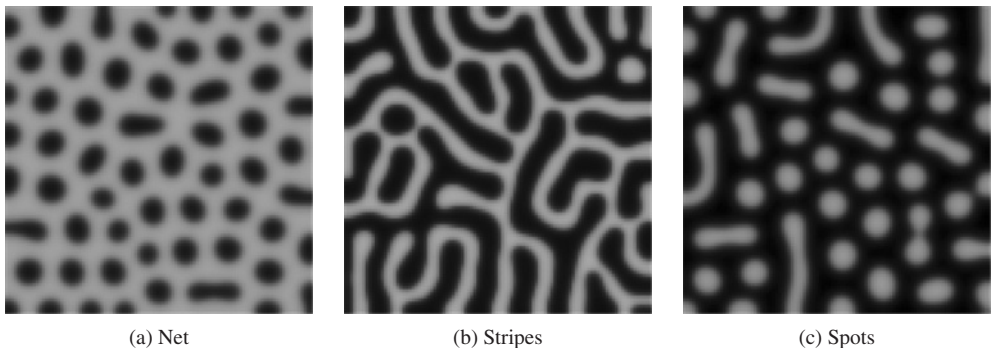


Figure 1.8 Patterns generated by reaction-diffusion equations resemble those found in nature on the fur of animals

method is able to reconfigure itself to maintain total coverage even when cameras are moved or rearranged.

Henderson *et al.* [16] use reaction–diffusion to form paths in a sensor network along which messages or agents move. A surface point corresponds to a node in the sensor network and the diffusion of chemical substances among neighboring points is accomplished by radio transmission. Without any centralized control the desired pattern is formed in the network where each sensor node only acts according to simple RD equations. A medium access control (MAC) for ad hoc networks based on reaction–diffusion is presented by Durvy and Thiran [10]. Global transmission patterns are set up only by local interactions among ad hoc nodes and a collision-free transmission pattern is set up. Furthermore, an efficient scheduling of the transmissions in multihop wireless networks with high traffic loads is possible.

One drawback in the applicability of reaction–diffusion is that it suffers from the same problems that were previously mentioned for all biological systems. In order to reach a stable pattern, reaction–diffusion requires time until convergence is reached. Therefore, it is not well suited for application in networks which have dynamically changing environments with fast-moving nodes.

1.4 Evolutionary and Adaptive Systems

So far we discussed that biological systems operate with feedback-based control loops to adapt to changes in the environment. This phenomenon can be seen in the natural selection process as well. Species evolve over many generations to be able to better survive in a dynamically changing habitat by adopting new features. Influences from mutation introduce diversity and adaptability to new challenges.

Genetic algorithms (GA) are a popular class of evolutionary algorithms as a heuristic method for the solution of optimization problems. While they are not necessarily related to networking problems, they are mentioned here because they nicely illustrate the operation of genetic evolution, which is a common principle in biologically inspired networks. In GA, a problem is encoded as (binary) individuals (*chromosomes*) and its evolution is performed over many generations applying the typical operations of natural selection, mutation and reproduction. Selection is the process where the fitness of individuals is evaluated upon which the new generation is selected. For the pool of selected individuals, parents are chosen which then produce a child by crossover and mutation, see Figure 1.9. In crossover, parts of the chromosomes of the parents are broken and recombined to form new different chromosomes. Mutation further adds to the diversity by modifying the genetic material. This process repeats itself until a termination condition

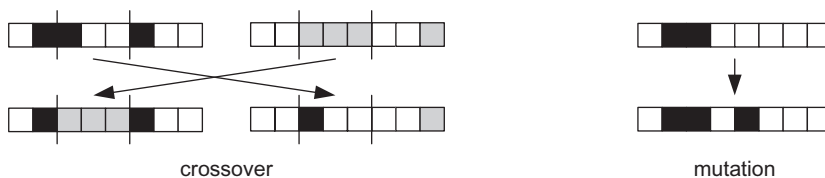


Figure 1.9 Reproduction by crossover of parental chromosomes and mutation

is reached. The resulting solution corresponds to the evolutionary adaptation of a species to its environment.

1.4.1 Models with Interacting Populations

While in genetic evolution the adaptation process is performed over the course of many generations, adaptive feedback models can also be used on a much smaller time scale to model the behavior of interacting populations. Interaction among species causes that the involved populations are affected by each other. In this section we discuss some approaches where models from mathematical biology are used in control algorithms for communication networks.

Basically, population models are used to investigate the dynamic interaction between different kinds of animal species. Murray [24] distinguishes three main types of interaction among populations depending on their growth rates:

- The populations are in *predator–prey* situation if the growth rate of one population is increased at the cost of that of the other.
- The populations are in *competition* if the growth rate for both populations is decreased.
- The populations are in *sympiosis* (or *mutualism*) if the growth rate for both populations is increased.

Although we speak of species, the models can be directly applied to other fields like communication network protocols. In this case, the population of a species would correspond to, for example, transmission rates.

1.4.1.1 Predator–Prey Models

Perhaps the most well-known example is the predator–prey model, which is also known as the *Lotka–Volterra model*. In general, the model is based on the *logistic population model*, which describes the evolution of the population over time and is bounded by a theoretical *carrying capacity*. The carrying capacity is the maximum population limit which can be supported by the environment. For a single population $N(t)$, growth rate $\varepsilon > 0$, and carrying capacity K , the logistic equation takes the following form.

$$\frac{dN}{dt} = \varepsilon \left(1 - \frac{N}{K} \right) N \quad (1.4)$$

The Lotka–Volterra model can be easily extended to n species, all influencing each other with a competition coefficient $\gamma < 1$, which we will assume to be equal for all species.

$$\frac{dN_i}{dt} = \varepsilon \left(1 - \frac{N_i + \gamma \sum_{j=1, j \neq i}^n N_j}{K} \right) N_i \quad (1.5)$$

It can be noted that when new species sequentially enter the system, the populations adapt in such a way that they converge to equal values among competing species. This

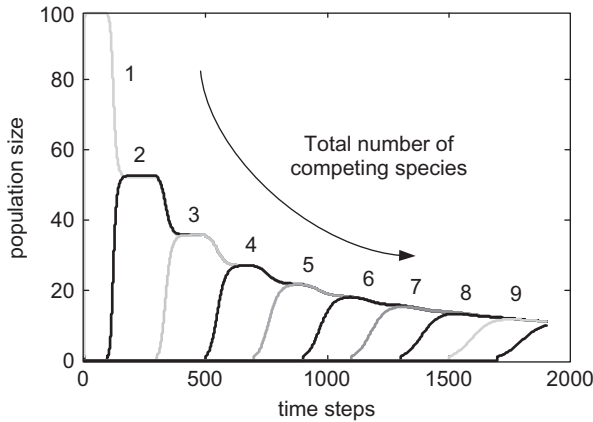


Figure 1.10 Competition model with 10 species that are successively added

feature can be used for example to control the adaptation rate of TCP's congestion control mechanism in a faster, fairer and more efficient way than by the *additive increase multiplicative decrease* (AIMD) approach currently found in TCP [15].

Figure 1.10 illustrates the changes in populations of 10 species when using Equation (1.5), where $K = 100$, $\varepsilon = 1.95$, and $\gamma = 0.90$. Each of the 10 species successively joins the system and their survival and convergence conditions are identical by using the same mutual influence factor γ for all of them. Even when two or more species exist, each independently utilizes Equation (1.5) to obtain N_i , and the population of the species converges to the value equally shared among all competing species. The changing population trends of the species depicted in Figure 1.10 are ideal for controlling the transmission speed of TCP. That is, by using Equation (1.5) for the congestion control algorithm of TCP, rapid and stable link utilization can be realized, whereas each TCP connection can behave independently as an autonomous distributed system. The population size of a species N_i corresponds to the transmission rate of a TCP connection i and K can be seen as the physical bandwidth of the bottleneck link. The method requires, however, knowledge of the transmission rates of all other connections sharing the bottleneck link represented by the sum over all other N_j . This is estimated by obtaining the available bandwidth on the bottleneck link through inline measurements. Unlike other TCP variants, the method using a multispecies Lotka–Volterra model for TCP transmission rate control does not show the typical sawtooth-shape when packet loss is encountered, but rather converges smoothly to the optimal solution. Simulation studies also indicate that the method shows higher scalability than other TCP variants and is therefore well suited to be applied as transport layer protocol for future high-speed networks.

1.4.2 Epidemic Diffusion

Another idea which originates from interacting population models in biology is the concept of using epidemic protocols for information diffusion [11]. Especially, in P2P networks, epidemic models can easily be applied due to their scalability, robustness, resilience to failure and ease of deployment. Epidemic algorithms are based on the spread of a

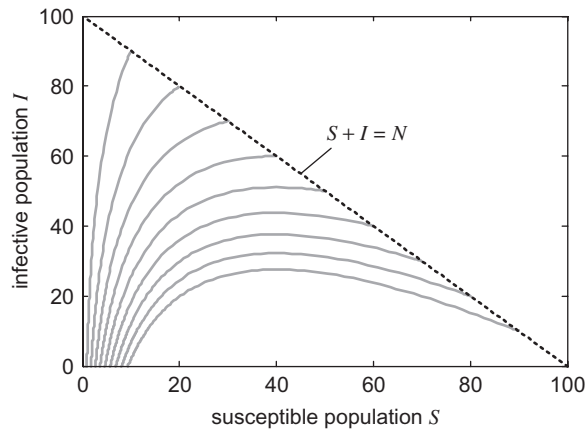


Figure 1.11 Trajectory plot of the SIR model in the $S-I$ phase plane

contagious disease where individuals of an initially healthy population of susceptibles become infected by those bearing the disease. After infection, they may themselves infect other susceptibles. The key aspect of this behavior is that the selection of a susceptible is entirely random and each individual has the same probability of becoming infected.

1.4.2.1 The SIR Model

Mathematical models are used to predict if a disease will turn into an epidemic or can be contained by vaccination [24]. Consider the outbreak of a disease in a closed system with a total population of N which remains constant, i.e., no individual enters or leaves the system. Then the population can be divided into three classes: the *susceptibles* S , who can be infected by the disease; the *infectives* I , who already have been infected and can transmit it to members of the susceptible class; and finally, there is the *removed class* R , who have either recovered from the disease, or become immune to it. This model is usually referred to as the *SIR model* and variations of it exist in the form of SI or SEIR, where a further class of individuals in which the disease is latent is added before being infectious.

Let $r > 0$ and $a > 0$ be constant parameters of the model corresponding to the *infection rate* and *removal rate* of infectives, respectively. Then, the transitions between the populations can be described by the following equation system.

$$\frac{dS}{dt} = -r S I \quad \frac{dI}{dt} = r S I - a I \quad \frac{dR}{dt} = a I \quad (1.6)$$

The mathematical formulation requires a set of initial conditions, given by the initial populations $S(0) > 0$, $I(0) > 0$ and $R(0) = 0$. If $S(0) < \rho = a/r$, which is called the *relative removal rate*, then $I(t)$ decreases and approaches 0 for $t \rightarrow \infty$, so the infection dies out. On the other hand, if $S(0) > \rho$, there is an epidemic since $I(t)$ initially increases. Due to this threshold behavior of $S(0)$ at ρ , we also speak of a *threshold problem*. The phase plot in the $S-I$ phase plane is shown in Figure 1.11 for $N = 100$, $a = 2$, and $r = 0.05$. Since $R(0) = 0$, all trajectories start on the line with $S + I = N$ and do not exceed this line.

Such types of models can also be used to characterize the diffusion properties of network protocols. For example, Leibnitz *et al.* [21] use a model based on SIR to predict the influence of pollution, i.e., maliciously injected contents, in P2P file-sharing networks. In a finite population of peers connected to a P2P network, several peers sharing a corrupt version of the file are assumed. The mathematical model includes the popularity of certain files and a detailed model of the downloading process with error checks and bandwidth sharing, as well as a user model which includes impatience. It was shown that a small number of malicious peers is sufficient to severely disrupt the file diffusion process. Studies investigating the file dissemination behavior can be used for designing more efficient content distribution systems using P2P networks.

1.4.2.2 Epidemic Methods in Communication Networking

A lot of theoretical research has been conducted on the mathematics of epidemics and many variants of epidemic information dissemination methods for communication networks have been studied. While mathematical models like the SIR model are used to predict how to contain a disease, the goal in epidemic diffusion is rather to ‘infect’ as many nodes as possible by sending information to them.

Basically, every node buffers the message it receives and forwards it a number of times to another node from a set of limited size, denoted as the *fan-out*. The outcome of the dissemination process is usually bimodal as indicated by the threshold behavior described in the previous section. Either almost all nodes receive the information at the end of the process or almost none. Eugster *et al.* [11] identify four key requirements on epidemic algorithms as:

- *Membership* – how nodes perceive neighboring nodes.
- *Network awareness* – how to make the connections between nodes reflect the actual network topology.
- *Buffer management* – which information to drop when the storage buffer of a node is full.
- *Message filtering* – how to make the nodes only filter information that is relevant to them.

The main advantage of epidemic algorithms is that they do not need to detect or reconfigure in the case of failures as they proactively choose the next hops when they are needed. Furthermore, the decision of forwarding a packet is made autonomously by each node, so if a single node fails, the system will nevertheless continue in operation.

While epidemic diffusion is used for distributing information to a large group of nodes, the basic principle can also be observed in *gossip*-based routing [14] in ad hoc networks. Due to the high degree of mobility and nondeterministic topology in ad hoc networks, the destination node must first be queried and a path set up to it. Usually, this is done by flooding, i.e., duplicating packets at each node and passing them on to each neighbor which has not yet processed this request. However, this causes a lot of overhead packets that can be reduced when the next hop node is chosen only with a certain probability.

1.5 Conclusion

In this chapter we discussed some basic principles and examples of biologically inspired networking. Methods from biology find increasing interest among the research community due to their attractive features like scalability and resilience to changes in the environment. By getting inspiration from biological systems, we can establish fully distributed and self-organizing networks. Especially networks which operate under no coordinating unit with highly autonomous nodes can profit from such methods. Furthermore, by utilizing the inherent fluctuations and noise in biologically inspired networks, they become more tolerant to perturbations, resulting in a greater stability and resilience.

However, we should also keep the following limitations in mind. Biologically inspired methods are often slower in reaction than conventional control algorithms. The reason for this lies in the way the adaptation is performed in nature. Often many thousand generations are needed for a species to evolve and adapt to a changing environment. Furthermore, adaptation is often done without knowing the target function, but simply by negative feedback. Individuals which have mutated to bear unfavorable features with disadvantages compared to other individuals will die out (*survival of the fittest*). This trial-and-error development is often driven by fluctuations and is thus not a directed development to a certain genetic feature. When biological methods are applied, it should be considered that they are in many cases useful heuristics which perform well for certain problem types, especially in the presence of incomplete or fuzzy input data, but may not be as good in performance as real optimization methods.

For this reason, we should be careful not to simply mimic biology, but to also keep a close eye on the problem for which a solution is sought. It is of great importance to thoroughly understand the specific features of the biological system before exploiting it in an interdisciplinary manner [29]. Therefore, after designing a new algorithm based on a biological method and its desirable features, it is necessary to move toward the application of the model and establish a more concrete mechanism with the detailed networking problem in mind. Not all self-adaptive approaches are equally applicable to all network types. However, we should certainly take inspiration from biology since nothing is more adaptive and resilient than nature.

References

- [1] Alberts, B., Johnson, A., Lewis, J. *et al.* (2002) *Molecular Biology of the Cell*, Garland Science, New York.
- [2] Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) *Swarm Intelligence: From Nature to Artificial Systems*, Oxford University Press, New York.
- [3] Carreras, I., Chlamtac, I., De Pellegrini, F., *et al.* (2007) BIONETS: bio-inspired networking for pervasive communication environments, *IEEE Transactions on Vehicular Technology*, **56**(1), 218–29.
- [4] Dasgupta, D. (ed.) (1999) *Artificial Immune Systems and Their Applications*, Springer, Berlin.
- [5] Di Caro, G. and Dorigo, M. (1998) Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, **9**, 317–65.
- [6] Di Caro, G., Ducatelle, F. and Gambardella, L.M. (2005) AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications (Special Issue on Self-organization in Mobile Networking)*, **16**(5), 443–55.
- [7] Dorigo, M. and Stützle, T. (2004) *Ant Colony Optimization*, MIT Press, Cambridge.
- [8] Dressler, F. (2005) Efficient and scalable communication in autonomous networking using bio-inspired mechanisms – an overview. *Informatica*, **29**(2), 183–8.

- [9] Dressler, F. (2006) Benefits of bio-inspired technologies for networked embedded systems: an overview. *Proceedings of Dagstuhl Seminar 06031 on Organic Computing – Controlled Emergence*, January, Schloss Dagstuhl, Wadern, Germany.
- [10] Durvy, M. and Thiran, P. (2005) Reaction-diffusion based transmission patterns for ad hoc networks. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, March, Miami, FL.
- [11] Eugster, P.T., Guerraoui, R., Kermarrec, A.-M. *et al.* (2004) Epidemic information dissemination in distributed systems. *IEEE Computer*, **37**(5), 60–7.
- [12] Forrest, S., Hofmeyr, S. and Somayaji, A. (1997) Computer immunology. *Communications of the ACM*, **40**(10), 88–96.
- [13] Garrett, S. (2005) How do we evaluate artificial immune systems? *Evolutionary Computation*, **13**(2), 145–78.
- [14] Haas, Z., Halpern, J. and Li, L. (2006) Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, **14**(3), 479–91.
- [15] Hasegawa, G. and Murata, M. (2006) TCP symbiosis: congestion control mechanisms of TCP based on Lotka–Volterra competition model. *Proceedings of the Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communication Systems (Inter-Perf)*, October, Pisa, Italy.
- [16] Henderson, T.C., Venkataraman, R. and Choikim, G. (2004) Reaction–diffusion patterns in smart sensor networks. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April, New Orleans, LA.
- [17] Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, **79**(8), 2554–8.
- [18] Kaneko, K. (2006) *Life: An Introduction to Complex Systems Biology*, Springer, Berlin.
- [19] Kashiwagi, A., Urabe, I., Kaneko, K. *et al.* (2006) Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS ONE*, **1**(1), e49.
- [20] Krüger, B. and Dressler, F. (2005) Molecular processes as a basis for autonomous networking. *IPSI Transactions on Advanced Research: Issues in Computer Science and Engineering*, **1**(1), 43–50.
- [21] Leibnitz, K., Hoßfeld, T., Wakamiya, N. *et al.* (2006) On pollution in eDonkey-like peer-to-peer file-sharing networks. *Proceedings of the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB)*, March, Nuremberg, Germany.
- [22] Leibnitz, K., Wakamiya, N. and Murata, M. (2006) Biologically inspired self-adaptive multi-path routing in overlay networks. *Communications of the ACM*, **49**(3), 62–7.
- [23] Mirollo, R.E. and Strogatz, S.H. (1990) Synchronization of pulse-coupled biological oscillators. *Society for Industrial and Applied Mathematics Journal on Applied Mathematics*, **25**(6), 1645–62.
- [24] Murray, J.D. (2002) *Mathematical Biology I: An Introduction*, Springer, Berlin.
- [25] Perkins, C. and Royer, E. (1999) Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing System and Applications*, February, New Orleans, LA.
- [26] Sarafjanovic, S. and Le Boudec, J.-Y. (2005) An artificial immune system for misbehavior detection in mobile ad-hoc networks with virtual thymus, clustering, danger signal and memory detectors. *International Journal of Unconventional Computing*, **1**(3), 221–54.
- [27] Suzuki, J. and Suda, T. (2005) A middleware platform for a biologically inspired network architecture supporting autonomous and adaptive applications. *IEEE Journal on Selected Areas in Communications*, **23**(2), 249–60.
- [28] Timmis, J., Knight, T., De Castro, L.N., *et al.* (2004) An overview of artificial immune systems. In R. Paton, H. Bolouri, M. Holcombe *et al.* (eds.), *Computation in Cells and Tissues: Perspectives and Tools for Thought*, Natural Computation Series, pp.51–86, Springer, Berlin.
- [29] Timmis, J., Amos, M., Banzhaf, W. *et al.* (2006) Going back to our roots: second generation biocomputing. *International Journal of Unconventional Computing*, **2**(4), 349–78.
- [30] Turing, A.M. (1952) The chemical basis of morphogenesis. *Philosophical Transaction of the Royal Society (London)*, B(237), 37–72.
- [31] Wakamiya, N. and Murata, M. (2005) Synchronization-based data gathering scheme for sensor networks. *IEICE Transactions on Communications (Special Issue on Ubiquitous Networks)*, E88-B(3), 873–1.

-
- [32] Wang, M. and Suda, T. (2001) The bio-networking architecture: a biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications. *Proceeding of the IEEE 2001 Symposium on Applications and the Internet (SAINT)*, January, San Diego, CA.
 - [33] Yoshida, A., Aoki, K. and Araki, S. (2005) Cooperative control based on reaction-diffusion equation for surveillance system. *Proceedings of the International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES)*, September, Melbourne, Australia.

