

# A Survey on Biologically Inspired Algorithms for Computer Networking

Chenyu Zheng, *Student Member, IEEE* and Douglas C. Sicker, *Senior Member, IEEE*

**Abstract**—**Biologically Inspired Algorithms (BIAs)**, processes that mimic how organisms solve problems, offer a number of attributes well suited to addressing challenges presented by future computer networking scenarios. Such future networks will require more scalable, adaptive and robust designs to address the dynamic changes and potential failures caused by high heterogeneity and large scale networks. A variety of biological algorithms demonstrate characteristics desirable to network design, and significant effort has been placed on analyzing and developing the corresponding BIAs and applying them to computer networking applications. This paper provides a comprehensive survey of BIAs for the computer networking field, in which different BIAs are organized and explored based on their: (1) biological source; (2) mathematical model; (3) major application; (4) advantages to corresponding “classic” approach; (5) limitations and border conditions; and (6) potential directions for future applications. The paper also compares performance amongst each type of BIA, and compares BIAs that are inspired by different biological sources but are applicable to similar networking applications. The paper concludes by offering a framework for understanding the application of BIAs to problems in the computer networking space.

**Index Terms**—self-organization, biologically inspired algorithms, computer networking, artificial immune systems, homeostasis, cellular signaling networks, chemotaxis and morphogenesis, division of labor, pattern formation, corpse clustering and brood sorting, foraging, pulse-coupled oscillator, information epidemics, biological symbiosis, predator-prey relationship

## I. INTRODUCTION

HERE is growing demand for networked systems, with consumers ever more dependent on Internet connectivity. However, communication devices vary in their functionality, mobility and resource constraints, while networks vary in their architectures. Applications often work in a large scale among different devices, which requires effective coordination, efficient resource utilization, robust failure handling, and adaptive system tuning amongst participating components. As a result, new designs in the computer networking field must be more scalable, adaptive and robust to provide effective and efficient services despite potential failures and dynamic changes caused by high heterogeneity and large scale networks.

Compared to the traditional central and static networking control mechanisms, Self-Organization (SO) is a promising approach for fulfilling current (and future) network design

Manuscript received November 2, 2011; revised May 12, 2012 and November 3, 2012.

C. Zheng and D. C. Sicker are with the Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA (e-mail: {chenyu.zheng, sicker}@colorado.edu).

Digital Object Identifier 10.1109/SURV.2013.010413.00175

requirements. Schemes from biological phenomena are appropriate candidates for SO-based network control, as they usually employ simple and sometimes identical operating rule sets to guide each individual to perform local observations and interactions with others to collaboratively produce effective and efficient operational patterns toward the goals of the entire group. For instance, in the problem of data forwarding in large scale wireless ad hoc networks, traditional flat routing schemes are not sufficiently scalable to provide effective services because of the overhead created by the large number of flooded control messages and the computation and memory requirement of large routing tables [1]. In contrast, SO-based hybrid hierarchical routing schemes, in which nearby nodes are grouped into clusters for reactive inter-cluster routing and proactive intra-cluster routing, could help boost scalability of the data forwarding protocol. SO-based clustering scheme makes the system adaptive and robust toward frequent topology changes. SO-based hybrid hierarchical routing also offers emergent behavior for data forwarding, where individual nodes perform local operations while the entire resulting infrastructure provides features, such as load-balancing, without explicit central control. Similar examples could be found in optimizing performance, reducing energy consumption, and enhancing the robustness of highly distributed heterogeneous systems such as data centers, grids, and clouds [2]. Because these systems are composed of a large number of distributed and heterogeneous entities, it is inefficient to tune the parameters of each entity through a centralized or a top-to-bottom hierarchical control. SO-based control has each entity individually tune its parameters to collaboratively implement system operational goals and enable quick decisions to react to failures of other entities, providing robustness to the entire system.

Additionally, biological phenomena show characteristics that are desirable for new network application designs, such as (1) *Immunity*: immune systems of vertebrate animals are able to block potentially harmful proteins from entering their bodies, are able to identify and destroy the potentially harmful proteins that enter their bodies, and are able to stay immune to the same types of potentially harmful proteins that have been terminated before; (2) *Homeostasis*: most biological systems are able to maintain a stable status by taking actions upon the observed continuous changes in their internal and external environments; (3) *Cellular signaling pathways*: cells of organisms can communicate with each other at the molecular level based on simple response rules toward different types of messages, and this communication pattern can lead to a self-organized, coordinated system behavior in an emergent way; (4) *Division of labor*: in societies of insects, such as ant

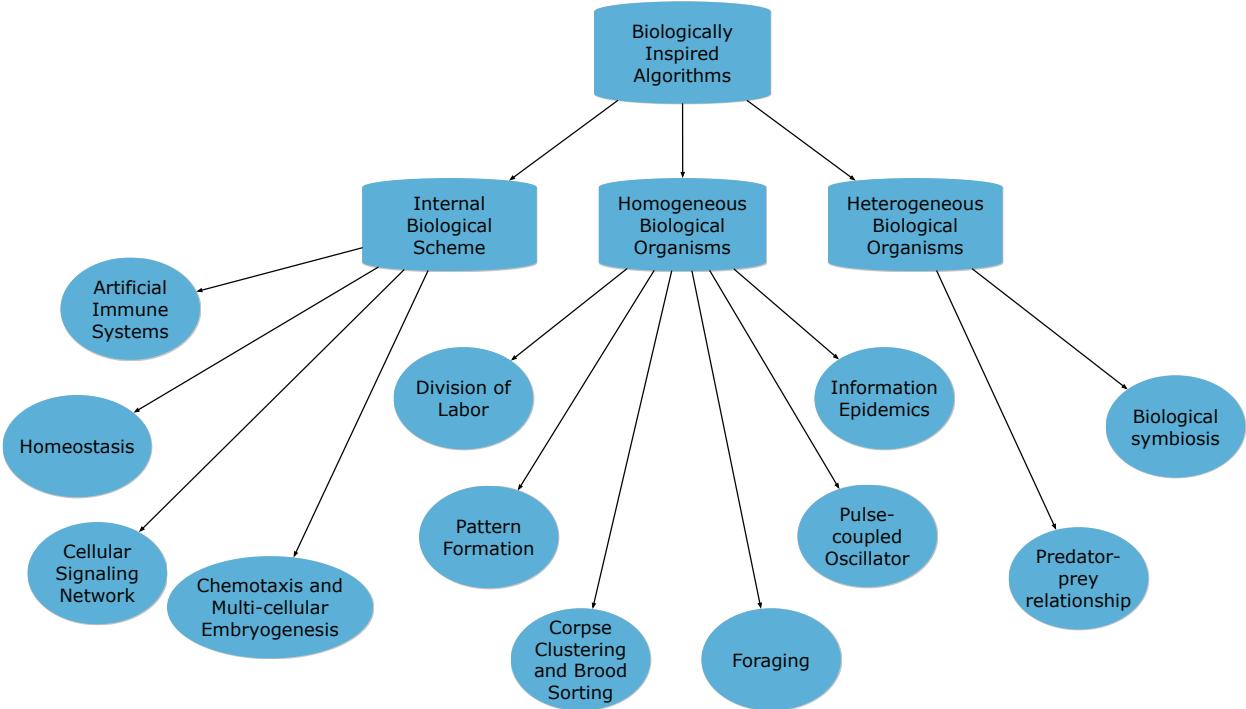


Fig. 1. A Taxonomy of Biologically Inspired Algorithms for Computer Networking

and honeybee colonies, different tasks are performed simultaneously by individuals that are better equipped for a given task, while the entire decision making process is executed distributively by each individual; (5) *Pattern formation*: various patterns on the surfaces (or in the structures of bodies of fish and mammals) are created simply based on the spatial distribution of special chemicals; (6) *Corpse clustering and brood sorting*: some insects are able to form “cemeteries” to clean the nest, and are able to gather the larvae according to their sizes, which is achieved by each individual insects performing pick up and drop off behaviors; (7) *Foraging*: insects are able to find the shortest paths to food sites through stigmergy behaviors, in which individuals indirectly communicate with each other through the environment, which they modify and react to; (8) *Synchronicity*: many biological systems show synchronized behaviors purely based on the local observation and adjustment of individuals; (9) *Chemotaxis and multi-cellular embryogenesis*: identical embryonic cells are able to differentiate to form different structures of the body, are able to produce spacing patterns throughout the organism, are able to repair wounds and regenerate themselves based on the ability of neighboring cells to detect regional death, according to the distribution of chemicals in their local environment, and are able to respond accordingly.

There also exist patterns and relationships among individuals in biological colonies that could be used to understand existing computer networking applications better and to guide their development. This includes (1) the spreading pattern of infectious diseases in *Epidemics* can be used to design data forwarding protocols for disconnected or intermittently connected networks; (2) the population relationship among different competing predators and their prey in *Predator-prey interactions* can be employed for coordinating nodes that

compete for limited resources in the network; and (3) the population relationship among coexisting bacteria in a shared medium in *Biological symbiosis* can be applied to coordinate the behavior of nodes that rely on each other in computer networks.

Research efforts have focused on analyzing these biological phenomena, developing the corresponding Biologically-Inspired Algorithms (BIAs) and applying them to computer networking applications. These applications are widely distributed in different layers of the network and different types of networks – from S-Net formed signal carrier or modulator for computable moving waves in the physical layer [3], to dynamic spectrum access schemes for cognitive radios in the MAC layer [4], up to the cooperation models for different types of peer-to-peer file-sharing networks in the application layer [5].

This paper offers a comprehensive survey of BIAs for the computer networking field. BIAs are organized and explored based on their: (1) biological source; (2) mathematical model; (3) major application; (4) advantages to corresponding “classic” approach; (5) limitations and border conditions; and (6) potential directions for future applications. The taxonomy of BIAs in figure 1 is selected based on the fact that most algorithms inspired from the same biological sources would naturally share similar mathematical models. Indeed, it is feasible to categorize the common characteristics and applicable scenarios of BIAs, along with their development trend, based on the groups of biologically inspiring sources.

To cluster the BIAs, we devised three major categories of biological inspiring phenomena:

- Those inspired by the *internal development, maintenance, and reaction schemes in organisms*, in which different components of organisms cooperate in order to achieve

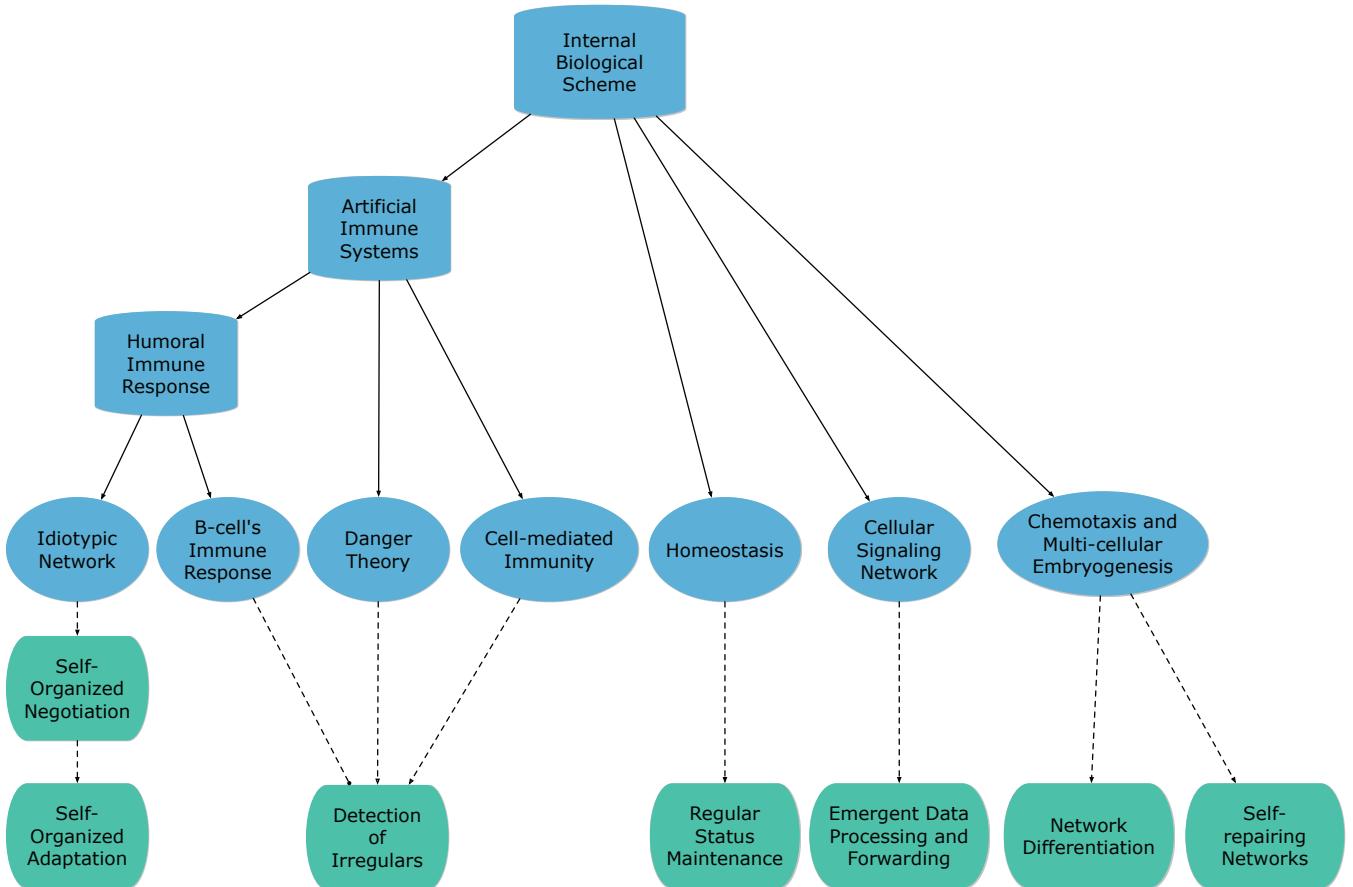


Fig. 2. A Taxonomy of Algorithms Inspired by Internal Biological Schemes (blue) with Their Applications (green)

different biological goals. The cooperation usually consists of interactions among internal components and the ones between internal components and external environments. These BIAs are summarized in figure 2.

- Those inspired by the *self-organized, emergent behaviors or patterns* that are performed or created by distributed homogeneous individuals from the same biological colonies. These BIAs are summarized in figure 4.
- Those inspired by the *interactions among distributed heterogeneous individuals from different colonies*, through which the dynamic balance amongst different colonies are maintained. These BIAs are summarized in figure 5.

The paper is organized as follows: Section 2 analyzes BIAs inspired by internal schemes in biological organisms; Section 3 analyzes BIAs inspired by the collective behaviors of homogeneous biological individuals; Section 4 analyzes BIAs inspired by the interactions among heterogeneous biological individuals. Finally, section 5 provides a discussion of our framework for applying BIAs to networking problems and section 6 concludes the paper.

## II. ALGORITHMS INSPIRED BY INTERNAL BIOLOGICAL SCHEMES

In this section, we review BIAs inspired by the internal development, maintenance, and reaction schemes in organisms. Basically, different components of organisms cooperate in order to achieve multiple biological goals. The cooperation generally consists of interactions among internal components,

along with interactions between internal components and external environments. The taxonomy of these algorithms is provided in figure 2.

As described below, these BIAs can be applied to network design in a straight forward fashion. There is usually no universally defined mathematical model for the procedure in each algorithm. Operating rules for components in a networked system are designed based on behaviors of corresponding components in biological systems. Table I summarizes networking applications that are suitable to employ algorithms inspired by internal biological schemes.

### A. Artificial Immune Systems

Artificial Immune Systems (AISs) inspired by the immune systems of vertebrate animals detect environmental changes or defend against abnormal behaviors in the computer system. Vertebrate Immune Systems (VISs) have multi-level defenses against *pathogens*, which are either non-self or potentially harmful proteins. These defenses include (1) physical barriers, e.g., skin, fur, and secretions, for preventing pathogens from entering the organism, (2) an innate immune system that uses *leukocytes* to non-specifically destroy or devour invading pathogens and potentially harmful cells, and (3) an adaptive immune system for rapid response to previously encountered pathogens using *lymphocytes*, such as B-cells and T-cells. There are three metaphors from VISs that most AISs employ:

TABLE I  
SUITABLE NETWORKING APPLICATIONS OF ALGORITHMS INSPIRED FROM INTERNAL BIOLOGICAL SCHEMES

BIAs	Application Fields
B-cell's Immune Response	
Cell-mediated Immunity	Detection of non-self or misbehaving individuals
Danger Theory	
Idiotypic Network	Self-organized negotiation on configurations or components
Homeostasis	Maintenance of system regular status
Cellular Signaling Network	Emergent data process and forwarding
Chemotaxis and Multi-cellular Embryogenesis	Differentiation of networking structure or functionalities; Self-repairing networks

humoral immune response (B-cell activity), cell-mediated immunity (T-cell activity), and danger theory [6], [7], [8].

1) *Humoral Immune Response*: In humoral immune response, B-cells identify pathogens by the *antigen* they produce in the innate immune phase. Based on this, they destroy the pathogens, and then secrete antibodies that stay inside an organism for a relatively long period, which immediately react to pathogens with the same antigen type.

The long memory and self-learning properties of this mechanism are adapted by AISs in the design of Intrusion Detection Systems (IDSs). In the IDS proposed by Kephart [9], decoy programs triggered by data integrity checking and activity monitoring examine suspicious programs and produce signature strings, like the antigen generated in the innate immune phase. Once compared with the signatures of uninfected programs, the signatures of confirmed viruses are inserted into the database of the current node and broadcasted to the databases of neighboring nodes as the paratope. The paratope is the part of an antibody that can identify and bind to a specific type of antigen. When the same type of potential malicious program invades the system again, the records in the database could be used to quickly identify them.

Research on idiotypic networks [10] suggested that antibodies are not isolated, but stimulate or suppress each other based on their idiootope. The idiootope is the part of an antibody that identifies types of other antibodies. Suzuki et al. [11] proposed a policy negotiation scheme for dynamically configuring policies based on the status of the system that applies this idiotypic network metaphor. Policies are modeled as antibodies, with the preconditions under which they are selected acting as paratopes. Policies stimulate or suppress each other's concentration to a degree, based on constraints between them. System conditions are modeled as antigens that stimulate the concentration of corresponding policies. The system determines the policy configuration based on concentrations. The resultant configuration feeds back the degree of stimulation or suppression among policies into reinforcement learning. This design produces a system whose configuration is dynamically and continuously adaptive to changing environments. It has been proven by the authors to surpass the static configuration system in the throughput metric under nominal operating conditions.

Atakan et al. [12], [13] proposed a distributed node and rate selection scheme for WSNs to reduce the number of designated nodes that are appropriate and responsible for transmitting sensing data to the sink. This scheme reduces

resource consumption and meets the event estimation distortion constraint of the sink. Each node is modeled as a B-cell, while the event source is modeled as a pathogen. The system tries to locate the minimum number of designated nodes as close to the event source as possible, and as far away from each other as possible. This is achieved by enabling each node to individually decide whether it should be a designated node or not, in order to obtain minimum distortion [14]. This decision making for each node is implemented by mimicking the stimulation and suppression of a B-cell in the B-cell stimulation model [15]. There are three actors for the decision making: (1) The affinity between the sensor node as a B-cell and the event source as a pathogen is the distance between the sensor node and the event source. (2) The affinity between the sensor node as a B-cell and its uncorrelated neighbor nodes as the stimulating B-cells is the sensor node's tendency of being a designated node if it is not in other designated nodes' application specific correlation radii. (3) The affinity between the sensor node as a B-cell and its correlated neighbor nodes as the suppressing B-cells is the suppressing influence of these neighbors on the sensor node's tendency of being a designated node. If the summation of the above three factors is greater than the preset threshold, the node will act as the designated node. The system also regulates the reporting frequency rate of each node according to the congestion in the forward path and the event estimation distortion at the sink.

Both Suzuki et al. [11] and Atakan et al. [12], [13] employed a common idiotypic network model as follows in eq. 1 and 2.

$$c_i(t) = \frac{1}{1 + \exp(0.5 - C_i(t))} \quad (1)$$

$$\frac{dC_i(t)}{dt} = \left( \frac{1}{N} \sum_{j=1}^N m_{ij} c_j(t) - \frac{1}{M} \sum_{k=i}^M m_{ik} c_k(t) + m_i - k_i \right) c_i(t) \quad (2)$$

$c_i$  denotes the concentration of the  $i$ -th antibody.  $N$  antibodies are suppressed by the  $i$ -th antibody, while  $M$  antibodies are stimulated by the  $i$ -th antibody.  $m_{ij}$  represents an affinity between antibodies  $i$  and  $j$ , while  $m_{ik}$  represents an affinity between antibodies  $i$  and  $k$ .  $m_i$  is an affinity between the  $i$ -th antibody and an antigen, while  $k_i$  denotes the natural extinction factor of antibody  $i$ . The control parameters in this model, i.e.,  $m_{ij}$ ,  $m_{ik}$ ,  $m_i$ , and  $k_i$  are configured based on the system status, along with relationships among different configurations.

The major criticism of the idiotypic network theory is its scalability [16]. Basically, each antibody needs to recognize

all antibodies that can either stimulate or suppress it. This requirement limits the network size. A system that contains vast configurations or components to negotiate cannot directly employ the idiotypic network theory because the number of antibodies needed is too large.

2) *Cell-mediated Immunity*: In cell-mediated immunity, T-cells bind and kill infected or potentially harmful cells. However, due to the pseudo-random generation of T-cell surface features used to bind target cells, some T-cells might destroy healthy cells as well. To prevent this side effect, T-cells are negatively selected before entering the blood circulation, i.e., all the T-cells whose surface features would bind to healthy cells are destroyed.

The random variation and negative selection schemes are adapted by AISs in the design of IDSs. Forrest et al. [17] implemented data integrity checking by using a repertoire of randomly generated strings, in which the strings from protected data are removed. If the potential verified strings match the ones in the repertoire, the system determines that there are verification failures.

LISYS [18], [19], proposed by Hofmeyr et al., detects intrusion in LAN by monitoring TCP SYN packets and identifying the non-self (potentially harmful) network connections using schemes, including negative selection. Here the connection information, i.e., IP addresses of source and destination and the TCP port, are compressed to single 49-bit strings that unambiguously represent that connection. The detection system randomly generates bit strings and exposes them to the environment for a “tolerization” period. If the randomly generated strings match the strings representing connections that occur frequently over a long period of time in the LAN, they will be eliminated. Otherwise, they become mature and any later match with a 49-bit connection string indicates that connection is non-self, i.e., not normally observed on the LAN and potentially harmful.

The negative selection concept has been applied to misbehavior detection applications as well. Le Boudec et al. [20] proposed an AIS to detect misbehaving nodes in the Dynamic Source Routing protocol (DSR) [21] routed MANETs, using negative selection schemes on the sequences of routing events: Each node’s sequence of packet headers is mapped to a compact representation of a string. The string representations are randomly generated by the system as detectors. Then, in an off-line learning phase, strings that match certified nodes’ packet header sequences are discarded. Finally, they are applied to match suspicious nodes’ string representations in the network. A node will be classified as misbehaving if the probability indicating it is suspicious surpasses the designed threshold over a large number of data sets.

Drozda et al. [22] summarized the advantages and disadvantages of the negative selection scheme as it is applied to misbehavior detection in WSNs: The algorithm itself is simple, and when the number of detectors to be produced is fixed and small, the algorithm can yield good experimental results [23]. However, the number of candidate detectors to the self set size must be exponential. In addition, there might be unfavorable holes in the non-self set that are undetectable for certain non-self set representations and matching rules [24] [25].

3) *Danger Theory*: The *danger theory* states that when cells are killed by pathogens, they emit a set of signals. The immune system is then able to acknowledge the existence and characteristics of the pathogens after receiving these danger signals.

It is inefficient, even impossible, to map the entire non-self universe to the detector set, particularly as the self and non-self set of the system will change over time. As a result, false positives on pathogen detection are produced in the negative selection procedure. The danger theory could be applied in AIS to overcome this disadvantage of the negative selection scheme [26], [27].

Aickelin et al. [28] demonstrate the advantages of danger theory as applied to IDS. In the proposed IDS, apoptotic signals corresponding to normal cell death represent consequences from legitimate actions that could be prerequisites for an attack. Necrotic signals corresponding to the death caused by pathogens represent the actual damage caused by a successful attack. The authors suggest that understanding the reaction scheme of VISs to mixed danger signals in danger theory could help the IDS concentrate on key types of danger signals and correlate the necrotic key types of signals to the apoptotic ones. The IDS must continuously update intrusion signatures and anomaly thresholds of pathogen detection in order to adapt to the changing network environment. This adjustment will affect the detection and in turn change the environment. By discovering how the overall immune responses in danger theory converge to an ideal status at a given time, the proposed IDS could be tuned to quickly converge to the most appropriate intrusion signature and anomaly threshold. In addition, inspired by danger theory, the proposed IDS tries to transmit danger signals from each node to its spatially, temporally, or logically nearest nodes, in order to detect ongoing intrusions.

Sarafijanovic et al. [23] added the danger signal scheme to their original design detecting misbehaving nodes in DSR routed MANETs [20] to achieve low false positives. A danger signal is generated by a sending node that does not receive the ACK for a packet. The signal, which contains the information of time and nodes correlated with this packet loss, is then sent over the route on which the packet loss took place, from the source node to all the nodes that can overhear it. Each node in the network determines one of its neighboring nodes as a misbehaving node if mature detectors generated from its negative selection procedure match relatively many of the compact representing strings of the neighbor’s packet header sequences. And additionally, if it receives danger signals that are related to these packet header sequences.

4) *Discussion*: All these three metaphors from VISs could be used to detect the non-self or misbehaving individuals in future networked systems. The metaphor of *humoral immune response*, i.e., B-cell’s immune response, has the advantage of fast detection of reemerged non-selfs, based on the match between features extracted from the detected individuals, along with features of previously encountered non-selfs in storage. However, the boost of detection speed is at the expense of storage consumption of extracted features of encountered non-selfs. It also suffers from a scalability problem on the huge number of features from the detected non-selfs. In addition, for

TABLE II  
PERFORMANCE COMPARISON AMONGST VIS INSPIRED BIAs

	B-cells Immune Response	Cell-mediated Immunity	Danger Theory
Storage-bound	Medium	High	Low
False Positives	None	High	Low
True Negatives	High	Medium	Low
Damage Precaution	Available	Available	Unavailable
Reemerged Irregular Detection	Fast	Fast	Fast
First Appearing Irregular Detection	Slow	Fast	Fast

the first appearing non-selfs, this metaphor could not provide fast detection.

On the contrary, the metaphor of *cell-mediated immunity* offers fast detection on both the first appearing non-selfs and the reemerged ones, based on the filter created by random variation and negative selection on regulars. But this metaphor requires more storage, since in order to yield good detection results, the size of the filtering set should be exponential to the size of the regulars. In addition, this scheme suffers from the problem of introducing false positives into the detection scheme because of the incomplete coverage over the non-self universe.

The metaphor of *danger theory* detects the non-selfs based on the signals sent by the regulars whose functionalities are affected by the non-selfs (danger signal). It provides faster detection on the first appearing non-selfs than the B-cell's immune response metaphor does. In addition, it requires less storage consumption on the predefined or detected danger signals than the cell-mediated immunity metaphor does. Moreover, the scheme, that detecting non-selfs based on the characteristics of their effects instead of their own characteristics, eliminates possible false positives of the the cell-mediated immunity metaphor. Along with these metrics, there is research space to improve the danger theory metaphor on the reaction scheme toward danger signals that are often mixed, so that key types of non-selfs could be identified and processed at a higher priority. The principles and details of how VISs deal with the mixed danger signal must be better understood for this improvement goal.

Garrett [29] evaluated the performance of the above three metaphors from VISs. The author pointed out that the B-cell's immune response and the *cell-mediated immunity* actually have similar symbols, expressions and processes. Indeed, both algorithms use a filtering set of non-self representations to identify non-selfs. The difference is how they generate this filtering set. *Humoral immune response* produces this set from previously encountered non-self entities, while *cell-mediated immunity* builds this set based on the behavior of self entities negatively selecting randomly generated representations. As a result, the former has the advantage of detecting reemerged non-selfs, while the latter has the advantage of detecting first appeared non-selfs.

From the performance point of view, the author stated that negative/positive selection algorithms can outperform Adaptive Resonance Theory based Artificial Neural Networks (ART ANN) [30] in terms of precision in detecting changes in a signal. However, their time complexity is greater than other classic algorithms that are used for anomaly detection.

The author also argued that compared to negative/positive selection algorithms, the danger theory can effectively minimize false positives. This is because negative/positive selection algorithms need to extract self-nonself information from many features that may change over time, while the danger theory only needs to make decisions on grounded dangerous and non-dangerous features, which are normally about the performance of the system and thus easy to evaluate. However, the danger theory suffers from the drawback that it has to wait until non-selfs damage the system. This makes it unable to take precautionary measures to defend non-selfs as negative/positive selection algorithms do. Table II summarized the performance comparison amongst the three VIS metaphors for non-self detection.

Another future application field of VIS BIAs is the problem of optimization for networked systems in continuously changing environments. This optimization scheme is based on the idiotypic network metaphor from the humoral immune response. The idiotypic network is a nonlinear dynamic system. Farmer et al. [31] demonstrated that it can provide sufficient richness in learning to fulfill system requests at the level of the Holland classifier system [32].

The systems suitable to employ this metaphor for optimization have the following common characteristics: In the system, there are different configurations or components, that aim for the same optimization goal, but have diverse effects, stimulating or suppressing each other to gain more control over the system. Changes in the environment stimulate or suppress a set of configurations or components. The metaphor enables different configurations or components to negotiate with each other based on stimulation and suppression relationships, so that a balance among them is achieved. As a result, the systems that apply this metaphor are continuously optimized to adapt to their changing environments. For this application, stimulation and suppression relationships among different configurations, along with definitions of relationships between changes in the environments and their corresponding configurations, must be correctly defined.

### B. Homeostasis

The phenomenon of biological systems maintaining their own stable status by reacting to sensed continuous changes in their internal and external environments is known as *homeostasis* [33]. This kind of phenomena can be observed in mechanisms like body temperature regulation or blood pressure maintenance.

Research efforts [34], [35] have been placed on homeostasis metaphors in nervous, endocrine and immune systems. Al-

though universally applied mathematical models or algorithms for homeostasis have not been defined, Neal et al. [36] proposed a model for homeostasis achieved by the cooperation between the Artificial Endocrine System (AES) and Artificial Neural Networks (ANN). In their model, different types of *gland cells* secrete corresponding *hormones* in respond to external stimuli. Concentrations of produced hormones will change due to geometric decay. As described by the following eq. 3, the traditional output  $u$  of an artificial neuron with  $nx$  dimensions,  $x$  inputs, and  $w$  weights, in ANN models is modified to include the hormone levels affecting the input weights  $w$  of the neuron:

$$u = \sum_{i=0}^{nx} w_i \cdot x_i \cdot \prod_{j=0}^{ng} C_j \cdot S_{ij} \cdot M_{ij} \quad (3)$$

$C_j$  is the concentration of hormone  $j$ ,  $S_{ij}$  is the sensitivity of receptor  $i$  to hormone  $j$ ,  $M_{ij}$  is the match between the receptor  $i$  of neuron input and the hormone  $j$ , and  $ng$  is the number of hormone types.

Besides this model, Owens et al. [37] proposed a general control architecture that maintains the homeostatic status of a system while ensuring that the utilities of the system have been fully utilized. In this architecture, *environmental sensors* are employed to measure the states of the system, which are represented in the form of *homeostatic variables*. Each homeostatic variable has a corresponding set point that defines the desired homeostatic state of the system. Utilities of the system are described as *tasks*, and the homeostatic responses based on the values of homeostatic variables are defined as *actuators*. The control scheme in the architecture employs actuators to keep the homeostatic variables at their set points while the system completes tasks. The paper pointed out that, in order to sensibly and tractably divide a system into these functional components, one must carefully consider the problem domain, the system purpose, the choice of the component mapping, and the design of the homeostatic control system. The design choices were well analyzed in the paper, but there is space for additional work in this area, such as providing precise mathematical models or algorithms for the control scheme of the original architecture. In addition, interaction schemes among different homeostatic control systems that mimic the interaction schemes among different biological homeostasis systems can be proposed.

The homeostasis metaphor has been applied to the field of security in operating systems. Somayaji et al. [38] introduced a Linux kernel extension that maintains homeostasis in the operating system's security in an autonomously responding fashion. Attacks are effectively stopped by monitoring active processes at the system-call level in real-time, and delaying (i.e., reducing the CPU cycles received by unusually behaving processes based on the monitored anomalies of system call sequences) or aborting the execution of compromised programs. At the same time, legitimate computation proceeds with minimal impact on performance. Twycross et al. [39] applied a similar benign response scheme to automatically prevent potential viruses from leaving the system by throttling the network connections of malicious programs.

As pointed out by Somayaji et al. [38], homeostasis systems tend to consume large amounts of computational resources

due to continuous observation, calculation and adjustment operations. Thus, before using this type of algorithm, one must carefully evaluate the tradeoff between maintaining regular system status and conserving system resources.

There is little literature that compares the performance of homeostasis algorithms with other approaches in terms of system status maintenance. Nevertheless, Owens et al. [37] proposed a performance metric that can be used individually to evaluate the acclimatization of homeostasis systems against changing environments. The authors define mutual information  $I$  [40], between the system  $S$  and the environment  $E$  as follows in eq. 4.

$$I(S : E) = H(S) - H(S|E) \quad (4)$$

$H(S)$  is the entropy of the system, while  $H(S|E)$  is the conditional entropy of the system with respect to the environment. Basically, the latter is the amount of system information that is not correlated to the environment. To evaluate the acclimatization of the system toward the changing environment, one needs to compare the rate of change in the environment to the rate of change of mutual information  $I(S : E)$ . If the former rate is relatively higher, one can infer that the system is overtrained to a specific environment. However, if the latter rate is significantly higher, one can then infer that the system cannot provide appropriate acclimatization against changing environments.

The metaphor of homeostasis can be applied to the problem of networked system status control in continuously changing environments in future research. The systems suitable to employ this metaphor have high priority in maintaining a stable status, guarantee sustainable functionality. The control mechanisms for the systems are the key components in these applications. They must be defined so that decisions on homeostatic responses based on changing environments can effectively adjust the systems toward a regular state. With future research on building more mathematical models for biological homeostasis systems, there will be more candidate mechanisms for the design of control systems.

### C. Cellular Signaling Network

At the molecular level, cells communicate with each other through a process called *signaling pathways*. In one mechanism, receptors located on the surface of the cell can bind a related molecule on the outside of the cell. The bound molecule activates signal molecules inside the cell, which can in turn activate further downstream signaling molecules, resulting in a protein-protein domino effect. Alternatively, molecules from remote cells traveling through the blood system could be bound by receptors within a cell. The resulting hormone-receptor complex could enter the nucleus of the cell and initiate gene transcription of the cell, which will trigger protein-DNA interactions to produce a cellular response [41], [42]. From a global point of view, each individual cell performs simple, but specific responses to information molecules, which depend on the type of the signal and the state of the cell, leading to a self-organized, coordinated system behavior in an emergent mechanism. This emergent property is useful for control applications on massively distributed systems. Although this phenomenon has not been mathematically modeled, there are two major applications that claim to be inspired

by the metaphor: (1) Rule-based Sensor Network (RSN) [43], [44] and (2) Fraglet [45], [46].

RSN is a light-weight, programmable control scheme for sensor and actor networks (SANETs). The information flow of the entire network is carried out by an architecture of data-centric forwarding, aggregation, and processing for self-describing messages. The role of messages in RSN is similar to that of information molecules in cellular signaling pathways. The local behavior of each node is directed by a rule interpreter in the form of simple state machines, which apply the rules in the node to received messages. Each rule is a combination of a condition predicate and a corresponding action of the node for that condition. All the messages that match rules' predicates are processed in an integrated way, in order to allow the RSN to perform data aggregation and fusion algorithms without maintaining a message history. Rules could be globally programmed so that the specific response of each node toward different types of messages could lead to a coordinated reaction on a higher organization platform. Indeed, RSN is shown to be able to perform collaborative sensing and processing in SANETs with purely local rule-based control. Along with the emergent self-organization property, RSN outperforms other SANET protocols for distributed sensing, data processing, and forwarding workloads with increased reactivity, scalability and reduced communication overhead. The key tuning parameter of an RSN is the execution period  $\Delta t$  of the rule interpreter, on which the rule interpreter periodically bases application of rules to the queue of the received message buffer. The larger the  $\Delta t$ , the better the data aggregation quality for RSN, since messages are stored longer in the buffer before being processed in an integrated way. However, the smaller the  $\Delta t$ , the lower the artificially introduced per hop delay for RSN. Dressler [47], [48] managed to maintain the optimal balance on the value of  $\Delta t$  using a nature-based feedback loop that was inspired by the promoter-inhibitor scheme of renin-angiotensin control for homeostasis of arterial blood pressure in mammals.

The main applications of RSN are data sensing and pre-processing in SANETs. Currently, there are few bio-inspired algorithms designed for a similar purpose. The reason it outperforms classic WSN control approaches [48] is that it distributes the data process task from traditional data sinks to each node. As a result, the amount of data to be transmitted is reduced, the scalability of the network is increased, and control instructions can reach actors faster. Nevertheless, these features are achieved at the expense of letting distributed entities converge to applicable solutions, which reduces the determinism of systems. From the viewpoint of a control mechanism, this can be a key disadvantage.

Tschudin [45] proposed a metabolic pathway inspired execution model similar to RSN for communication protocols, in which code and data are unified in the form of single units called fraglets. Nodes in the network process and forward incoming fraglets, which are symbol strings  $[s_1 : s_2 : \dots : s_n]$  encoded as packets, and represent data and protocol logic to accomplish communication and automatic network adaptation and optimization. In each node, there is a fraglet stored in the form of a multiset, to which incoming fraglets are added, and from which stored fraglets are continuously identified and processed. The identification and processing system of fraglets

belongs to the category of string rewriting systems. Fraglets are identified based on their front symbols and are rewritten and spliced into new ones based on the preprogrammed rules in each node's instruction set. The resulting fraglets are either added to local nodes' stores or forwarded to stores of other nodes. The instruction set consists of *transformation* rules, according to which single fraglets are processed, along with *reaction* rules, according to which two fraglets are jointly processed. The paper provided effective instruction sets for implementing a confirmed-delivery protocol and an  $N$  window size reordering flow control protocol.

Various research efforts have focused on improving the fraglet system. Yamamoto et al. [49] extended the original instruction set to support fraglets that build programs that can self-replicate, self-modify, and self-reproduce during their execution from simple quines up to genetic operators. Martinelli and Petrocchi considered applying security to fraglets, including following aspects: (1) extending the original instruction set with primitives for symmetric encryption and decryption in order to support security communication protocols among nodes [50], (2) adding access control mechanisms to the framework so that the send rule in the instructions set will be executed only if guarded, i.e., a fraglet could not be sent to a store unless it passes the check points defined by that store [51], and (3) building a role-based trust management frame work for the fraglet system in which digitally signed, quantitative trust credentials could be combined with proposed operators [52].

As demonstrated by Tschudin [45], the fraglet is capable of effectively defining communication protocols, and it has advantages of having less syntactic constraints and smaller size than other programming languages. However, both its control logic and fitness function must be manually defined and their functionalities are hard to verify analytically. Although Yamamoto et al. [49] showed that letting fraglet support intrinsic genetic programming can enable the system to self-modify, the generation process may influence the execution. In addition, invalid and seldom used instruction rules might be produced.

The metaphor of cellular signaling networks is suitable for future networked systems that heavily rely on information exchange and modification among simple rule controlled components to obtain emergent behaviors. These emergent patterns are produced to provide collaborative data processing and transmission control on top of the network layer in continuously changing environments, such as data integration and fusion, confirmed delivery, and flow and congestion control. Information coded in different forms in various systems is the only input and output. The output is generated from both the received information and the status of each component based on its simple interpreting rules. These simple rules could be either predefined or self-modified, which mimics the cell DNA evolution and behavior control scheme. The self-modified rule definition also provides an opportunity for applying quine programs in future research.

#### D. Chemotaxis and Multi-cellular Embryogenesis

The phenomenon of chemotaxis, that cells move according to the gradient of chemicals in the environment, and the ones

TABLE III  
EVALUATION ON BIOLOGICALLY INSPIRED ALGORITHMS

BIAs	Merits (advantages to classic approaches)	Flaws (limitations & border conditions)
B-cell's Immune Response	Fast detection on the reemerged non-selfs	Slow detection on first appeared non-selfs; Low scalability
Cell-mediated Immunity	Fast detection on both first appeared and reemerged non-selfs	The size of filtering set should be exponential to the size of regulars; Producing false positives due to incomplete coverage over the non-self universe
Danger Theory	Fast detection on non-selfs; Requiring small amount of storage	Requiring non-selfs to make damage before detection
Idiotypic Network	Broad utility for self-learning	Low scalability
Homeostasis	Stable system status	Overhead from continuous observation and computation
Rule-based Sensor Network	High scalability; Long network lifetime; Real-time response	Reducing the determinism of systems
Fraglet	Almost no syntactic constraints; Size of the fraglet is small	Need to manually define control logic and fitness function; Instruction rules are hard to analytically verify; Self-modification may block normal execution flow and introduce invalid and seldom used rules
Chemotaxis and morphogenesis	Robust; High scalability	High communication overhead
Division of Labor	Adaptive to status of tasks and individuals; High responsiveness; Low communication overhead	Low scalability to the number of tasks
Pattern Formation	Scalable to the size of networks; Robust to failures of nodes; Adaptive to changes in networks	High communication overheads for updating concentrations of activators and inhibitors
Corpse Clustering and Brood Sorting	Automatic determination of the number of clusters; Robust to outliers in data; Self-organized and decentralized; Independent of prior information	Slow when clustering low dimensionality data
Foraging	Robust and adaptive; Efficient resource utilization; High Scalability	High communication overhead from control messages
Pulse-Coupled Oscillator	Self-organized; High Scalability; Robust to fragile communication	Accuracy is affected by: estimation on delays, physical clock drift, and impacts from high priority processes
Epidemic Routing	Independent of simultaneous end-to-end paths; Robust to incomplete knowledge about the network	High latency; Extra overhead from multiple copies of data packets
Predator-prey Relationship	Smooth adaptation; High fairness	Stability of system equilibrium depends on competition rates of species
Biological Symbiosis	Effective resource coordination	Limited ability to represent competing relationships amongst complex systems

observed in multi-cellular embryogenesis, i.e., morphogenesis, offer biological primitives for self-organization in engineering systems. Nagpal [53] summarized these primitives and proposed the corresponding engineering implementations, most of which are morphogen gradient based schemes.

First, identical cells of embryos are able to differentiate to form different structures of the body based on their relative distances to each other. In the embryo development phase, certain numbers of cells first produce different morphogens, which disperse throughout the entire embryo. Each cell then

detects local values of different morphogens to determine its relative location to the different morphogen source cells. Each cell is triggered to form its target structure, i.e., it differentiates. This process could be performed recursively. The entire embryo first divides into a number of different cell groups. Each group then differentiates into different smaller cell groups. This subdividing differentiation process continues until the entire creature could be built with sufficiently differentiated cells. This metaphor could be applied to the formation of topology change aware networks. Nodes in the network

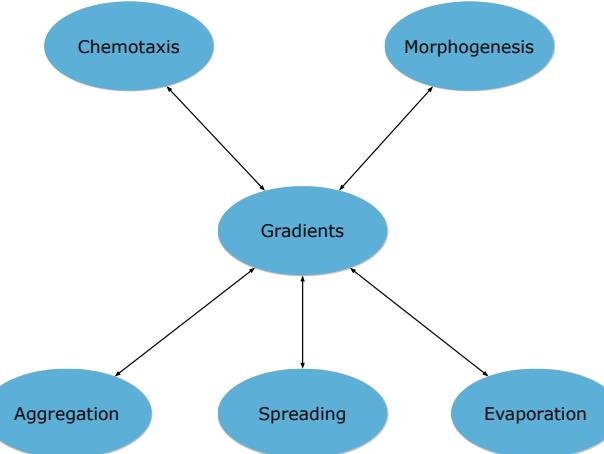


Fig. 3. Relationship of Gradient Models

that act as morphogen source cells continuously flood their own time-decaying morphogens, in forms of messages, to the entire network. Nodes receiving these messages could then determine their relative distance among each other according to their local morphogen values. If a node joins or leaves the network, its unique morphogen will show up or disappear because of time decay effects. The other nodes could then acknowledge the change of the network topology according to the changed local morphogen value.

Secondly, cells are able to produce unique spacing patterns throughout the organism. In each small region, the cells, which either are chosen or successfully compete to be differentiated, continuously produce their own morphogens within a small range. These morphogens inhibit the cells within this range from differentiating and keep the cells that are out of this range competing for differentiation. The pattern that no two differentiated cells will be within a small distance, and inhibited cells will be close to differentiated ones is generated over time. This metaphor could be used for topology formation of cluster-based, semi-structured, peer-to-peer networks where the super nodes of clusters act as differentiated cells. The network topology of nodes is formed just as the spacing pattern of cells in the organism based on the morphogen schemes.

Thirdly, cells are able to follow the gradient of a diffusing substance. This is achieved by each cell comparing its own morphogen value to the ones of its neighbors, along with moving toward the ones whose morphogen values represent that they are closer to the source. By applying this metaphor, connection-guaranteed routing in large regional failure scenarios could be achieved.

Lastly, organisms are capable of repairing wounds and regenerating themselves, based on the ability of neighboring cells to detect regional death and then respond appropriately. Each cell in the organism produces morphogens and expects certain gradients of morphogens from its neighboring cells. If it does not receive normal morphogen values from one of its neighbors, it will try to recreate or recruit a new neighboring cell with correct morphogen gradients. This process repeats until all the cells are satisfied with their neighbors. This scheme could be applied to create self-repairing structures

in computer networks by enabling functioning components to continuously diffuse and detect morphogen signals, along with replacing components of neighbors that are no longer able to produce desired signals.

Fernandez-Marquez et al. [54] explored the relationships between different gradient-inspired models. As shown in figure 3, three layers of patterns were analyzed in the paper. Both chemotaxis and morphogenesis algorithms belong to the top layer, which is an extension of gradient-based patterns. The gradient inspired algorithms are based on aggregation, spreading, and evaporation models as described in the following eq. 5.  $x_i$  represents the local morphogen concentration from  $i$ -th morphogen source.  $\alpha_i$  denotes the concentration of morphogen  $i$  changes due to evaporation. Function  $op$  models the aggregation process when multiple types of morphogens are locally present.

$$\begin{aligned} & \text{recv}\{x_1, \dots, x_n\} \longrightarrow \\ & \quad \text{send}(op(x_1 * \alpha_1, \dots, x_n * \alpha_n), \text{neighbors}) \end{aligned} \quad (5)$$

Chemotaxis and morphogenesis patterns are modeled as follows in eq. 6 and 7. Entities that are using chemotaxis algorithms follow the aggression of morphogens. Entities that are using morphogenesis algorithms determine their modality states, which are usually their locations toward morphogen sources, based on obtained morphogens.

$$\begin{aligned} & \text{morphogenesis} :: \{x_1, \dots, x_n\} \longrightarrow \\ & \quad \text{modality} = cp(x_1, \dots, x_n) \end{aligned} \quad (6)$$

$$\begin{aligned} & \text{chemotaxis} :: \{(neigh_1, x_1), \dots, (neigh_n, x_n)\} \longrightarrow \\ & \quad \text{follow}(neigh_i) \quad \text{if } x_i = op(x_1, \dots, x_n) \end{aligned} \quad (7)$$

Although chemotaxis and morphogenesis inspired algorithms have not been widely applied to computer networking problems, one can still infer its limitation in applications. This type of algorithm requires frequent exchange of morphogen information, which adds communication overhead.

The metaphor of chemotaxis and multi-cellular embryogenesis could be applied to role-influenced based schemes, in order to obtain differentiated architectures, structural and behavioral patterns, and self-repair properties for future networked system control. The systems suitable to employ this metaphor have a large number of components that must be differentiated into different operating roles. The design of the morphogen-based role differentiation scheme is critical to the success of these applications. The differentiated components take the corresponding workloads of their roles and spread their influence over the entire system to affect the role formation of other components.

### E. Summary

As shown by figure 2, there are several different BIAs that are inspired by the vertebrate immune metaphor and target the same goal of detecting non-selfs in networked systems. As a result, we are able to make a performance comparison amongst them in table II. However, for algorithms that are inspired by other internal biological schemes, there are few papers that

compare their performance with other BIA approaches. We argue it is possible that they are so highly differentiated to accomplish their service goals that no similar BIA schemes are available for the performance comparison. Nevertheless, the performance of this type of BIA is evaluated with their advantages to classic approaches and their limitations and border conditions in table III.

### III. ALGORITHMS INSPIRED BY HOMOGENEOUS BIOLOGICAL ORGANISMS

In this section, we review BIAs inspired by the self-organized, emergent behaviors or patterns that are performed or created by distributed homogeneous individuals from the same biological colonies. The taxonomy of these algorithms is provided in figure 4.

Basically, applicable mathematical models for this type of BIA are well defined. To apply them to computer networking, the network systems are required to possess a large number of homogeneous components with simple or identical operating rule sets. These components interact with each other, based on their local observations, to produce collaborative and effective behaviors. Table IV summarizes networking applications that are suitable to employ algorithms inspired by homogeneous biological organisms.

#### A. Division of Labor

Phenomena of division of labor have been widely observed in the task allocation of insect societies, such as ant and honeybee colonies [55], [56], [57], [58], [59], [60].

The mathematical model abstracted from these phenomena is the response threshold algorithm [61], which states that different tasks are performed simultaneously by the individuals that are best equipped for the given task. In other words, it tries to provide an optimal distribution of the work force among different workloads by letting each individual judge whether it should respond to a certain task or not. In the model, each task  $j$  has a *stimuli*  $s_j$  which is defined as the intensity of an activator associated with it. For each individual  $i$ , each task  $j$  has a *response threshold*  $\theta_{ij}$ , which refers to the likelihood of reacting to a task-associated stimuli. Individuals perform the task when the level of the stimuli  $s_j$  exceeds their threshold  $\theta_{ij}$ . The probability  $T_{ij}$  of the individual  $i$  engaging in task  $j$  is defined in the following equation, where  $n > 1$  determines the *steepness* of the threshold [4]. From the following eq. 8, the fact can be found that the task is likely performed when  $s_j \gg \theta_{ij}$  and unlikely when  $s_j \ll \theta_{ij}$ .

$$T_{ij} = \frac{s_j^n}{s_j^n + \theta_{ij}^n} \quad (8)$$

There are three advantages of the response threshold model for task allocation:

- The *stimuli* is determined from both the status of the associated task within the environment and the target of this task. The response threshold is determined from both the status of the individual within the environment and the object of the individual. As a result, changes of both the task and the individual can quickly be taken into consideration in the division of labor.

- The model is responsive in terms of its ability to converge when confronted with changes. To ensure quick convergence, intelligent choices in the threshold  $\theta_{ij}$  and its steepness parameter  $n$  should be made.
- From the individual's point of view, a consensus is made by individuals who are engaged in the same tasks at the same time. This consensus is achieved purely by the individual's own judgement, without communication amongst each other. This reduces the cost of communication and preserves the privacy of each individual. Indeed, along with the application for optimal task allocation, the response threshold algorithm has also been a competent candidate for distributed consensus development applications.

However, the response threshold algorithm has the disadvantage of low scalability to the number of tasks. The decision making on the entire colony's distribution of work force among different workloads is actually distributed to each individual's decision in response to each task. In other words, each individual has to evaluate its predisposition for all tasks in the environment before performing any action. This will introduce huge computation overhead when the workload set is large.

Although the *stimuli* and threshold could change over time due to the dynamic nature of the environment and the status change of both tasks and individuals, the dynamic of the threshold from reinforcement learning and the aging for tasks' stimuli are still considered to be necessary to the original response threshold model.

G. Theraulaz et al. proposed the reinforcement learning process whereby a threshold of an individual for a task decreases when the task is performed, and increases otherwise. The discrete form of this process within  $\Delta t$  is given by the following eq. 9 [62]:  $x_{ij}$  is the fraction of time spent by individual  $i$  engaging in task  $j$  within  $\Delta t$ .  $\xi$  and  $\varphi$  are the coefficients that describe the effects of learning and forgetting.

$$\theta_{ij} \rightarrow \theta_{ij} - x_{ij}\xi\Delta t + (1 - x_{ij})\varphi\Delta t \quad (9)$$

With  $x_{ij}$  being replaced by the probability  $p$  that individual  $i$  engaging in task  $j$  abandons the task per unit time, the above formula could be converted to a continuous-time formulation. The paper proved that the reinforcement on individuals' thresholds could give rise to specialized individuals from initially identical ones, even if the dynamics of environment and status of both individuals and tasks are not considered.

Table V summarizes the performance differences among division of labor models that are independent of information flow about task opportunity and individual availability. Currently, there is no quantitative performance study to compare the degree of optimization on work force distribution amongst these models. As argued by Besher et al. [63], these models are explanatory hypotheses that try to explain the colony level patterns of division of labor in the nature from different points of view. The authors further suggested that since they target different components of the processes that generate division of labor, it is promising to integrate these models to collaboratively achieve better results.

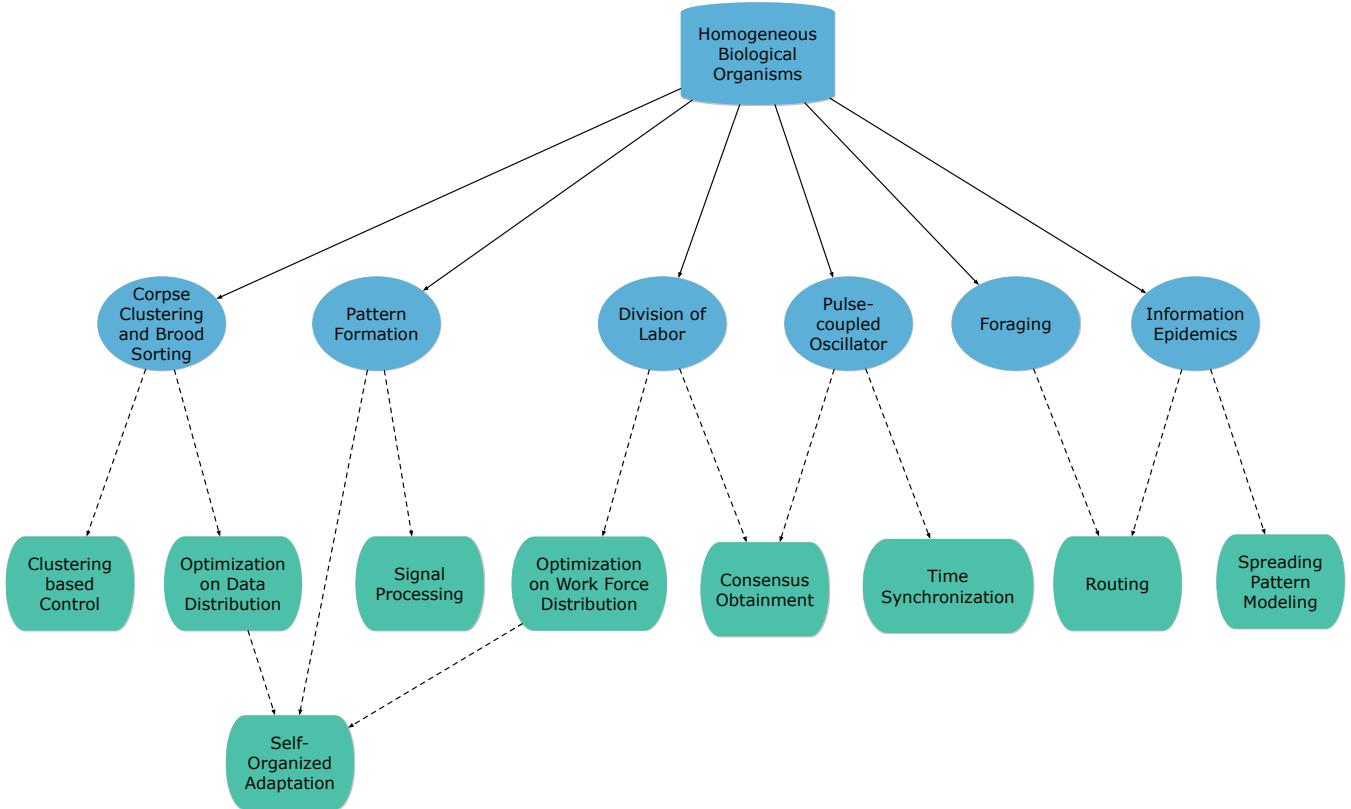


Fig. 4. A Taxonomy of Algorithms Inspired by Homogeneous Biological Organisms (blue) with Their Applications (green)

TABLE IV  
SUITABLE NETWORKING APPLICATIONS OF ALGORITHMS INSPIRED FROM HOMOGENEOUS BIOLOGICAL ORGANISMS

Biological Phenomena	Mathematical Models	Application Fields
Division of Labor	Response threshold algorithms	Optimization on work force distribution for workloads; Achieving consensus among components
Pattern Formation	Activator inhibitor algorithms	Self-organized system adaptation; Signal encoding, carrying, and modulation
Corpse Clustering and Brood Sorting	BM and SACA models	Clustering based network control; Optimization on data distribution among components
Foraging	ACO, and foragers and scouts models	Routing
Pulse-coupled Oscillator	PCO models	Self-organized time synchronization or desynchronization; Achieving consensus among components
Information Epidemics	SIR, SIRS, SI, and SIS models	Routing; Modeling the spreading patterns of viruses or information

TABLE V  
PERFORMANCE COMPARISON AMONGST DIVISION OF LABOR BIAs

	Response Threshold Model	Self-Reinforcement Model
Stimuli Change	When tasks are being performed	
Threshold Change	Based on internal status	
Temporal Polytheism	No	Yes
Learning Factor Change	None	When tasks have been performed successfully
Development of Specialists	Low; Variation in thresholds are amplified in task preferences	High; Low threshold individuals may continuously perform low stimuli tasks

The applications of the response threshold model lays in two aspects – *pursuing the optimal work force distribution* among workloads, and *achieving consensus* among different individuals.

For the first aspect, Labella and Dressler directly applied the model to the task assignment and selection in SANETs [64]. They changed the response probability so that the stimuli from other tasks are used instead of the threshold in the original ratio. Low et al. applied the response threshold algorithm to the sensor coverage problem. The scheme produces a self-organizing distribution of mobile sensors that have limited monitoring range to track the unpredictable moving targets in a dynamic environment [65]. Each region is considered as a task. The stimuli for it is defined as the ratio of the average delay for encountering other sensors and the average delay for encountering targets to be tracked. Sensors in the same region continuously compete with ratios and the number of observed targets. The winning sensor increases its corresponding threshold for staying in the region, and decreases otherwise. The scheme provides better coverage and more flexible response to the dynamic environment than either the classic static sensor placement scheme, or the auction based negotiation tracking strategy.

For the second aspect, Atakan and Akan targeted the problem of efficient and fair *spectrum sharing* of the available spectrum bands among coexisting cognitive radios, and proposed the BIOlogically-inspired Spectrum Sharing (BIOSS) algorithm [4]. In BIOSS, each secondary user could distributively determine the appropriate channels over which it could communicate with its peers without any coordination with other users. Secondary users who sit in the same channel achieve consensus among each other based on the response threshold model as defined in the following eq. 10. Each secondary user cognitive radio  $i$  is considered an individual, and each available channel  $j$  is considered a task. The estimated permissible power  $P_j$  [66] to the channel  $j$  is mapped as the task-associated stimuli. The required transmission power  $p_{ij}$  for cognitive radio  $i$  communicating through channel  $j$  is mapped as the response threshold. In this way, the channel having more interference can be preferred by the cognitive radio that needs less transmission power and vice versa. The thresholds are dynamic with the environment and the radio's own status.

$$T_{ij}^{csp} = \frac{P_j^n}{P_j^n + \alpha p_{ij}^n + \beta L_{ij}^n} \quad (10)$$

In addition, a reinforcement learning process is added to the original response threshold model. Secondary user cognitive radio  $i$  determines the threshold  $p_{ij}$  according to user requirements, acceptable interference level, acceptable path loss, along with channel link error, through the effective *spectrum management* model. As shown in the above formula, if channel  $j$  meets the QoS requirements for cognitive radio  $i$ , the learning factor  $L_{ij}$  in the denominator of the response probability is reduced by the learning coefficient  $\xi_0$ . Otherwise, the learning factor is added with the forgetting coefficient  $\xi_1$ . With this mapping design, BIOSS enables cognitive radios to easily adapt to the changing user requirements, along with the in-

stantaneous changes in the environment, such as small holding time, user mobility, and encountering the primary users. Thus, the system could distributively share the available spectrum bands without any central controller or any coordination.

The metaphor of division of labor could be applied to future networked systems that pursue optimal work force distribution among workloads, or achieving consensus among their components. The associated response threshold algorithm enables each component to independently choose to perform the tasks that fit its own capability for the system's adaptation goal. The optimization on work force distribution over workloads in the system is thus achieved through self-organization and consensus among the components that select the same tasks.

### B. Pattern Formation

Patterns on the surface or structure of fish and mammals' bodies are determined by the spatial distribution of chemicals called *morphogens*.

This pattern formation procedure was first modeled by Turing [67], in which, at each location of the system, two types of morphogens, namely the *activators* and the *inhibitors*, react with each other and diffuse along neighboring locations over time. This continuous reaction-diffusion mechanism will change the density of both morphogens at each location, and in turn will generate different patterns. Equilibrium (certain stable patterns) could be eventually obtained in the reaction-diffusion model. The equilibrium could be broken by random disturbances from both inside and outside the system.

The common mathematical model of reaction-diffusion mechanisms, which has been adapted by most biologically inspired applications, is in the form of second order partial differential equations as shown in the following eq. 11 and 12 [68]:  $u$  and  $v$  are concentrations of the activator and the inhibitor;  $F$  and  $G$  are nonlinear functions that represent the reaction between the activator and the inhibitor at current locations;  $D_u$  and  $D_v$  are the diffusion rates of the activator and the inhibitor; and  $\nabla^2$  is the Laplacian operator. The subtractors on the right hand side represent the diffusion of the activator and the inhibitor at current locations and effects from the neighboring locations.

$$\frac{\partial u}{\partial t} = F(u, v) - D_u \nabla^2 u \quad (11)$$

$$\frac{\partial v}{\partial t} = G(u, v) - D_v \nabla^2 v \quad (12)$$

The reaction-diffusion model has been proven to be effective compared to classic approaches for distributed and self-organizing control mechanisms of Wireless Sensor Networks (WSNs) under energy consumption and lifetime constraints in both simulations and experiments [69], [70], [71]. This model has been modified for different applications. For example, Hyodo et al. [72] sought to reduce the communication overhead and the surveillance video data in wireless camera sensor networks by applying the reaction-diffusion model to distributively adjust the video coding rate of each camera node in accordance to morphogen concentrations. A spot pattern of the video coding rate is formed by taking the directional and attenuated stimuli triggered by monitored moving objects in

the original model. Yoshida et al. [73] focused on decreasing blind spots in the surveillance area of plural camera networks by tuning the three parameters (pan, tile, and zoom) of the camera at each node. The paper uses a modified reaction-diffusion model, which integrates the guidelines for tuning in the form of evaluation functions. Durvy et al. [74] created a transmission pattern for the spatial reusability of media access control in multi-hop ad hoc networks. This pattern inhibits other transmissions in each node's immediate surrounding and encourages nodes located further away to transmit. The paper applies the reaction-diffusion model to control the Media Access Probability (MAP) of each node. Based on local observation, the diffusion of morphogens (i.e., MAP values) from far away nodes will increase the MAP of the current node, while the one from nearby nodes will decrease this probability.

Research on Smart Sensor Networks (S-NETS), which was first proposed by Chen et al. [75] and then carried out by Henderson et al. [3], not only employed the reaction-diffusion model to control the parameter tuning of smart sensor networks as the above applications, but also suggested that patterns generated by the S-NETS are capable of supporting other high-level algorithms or activities. This includes:

- Stripe, spot, or ring patterns to serve as encoders for both physical purposes, such as recording the paths of robots, along with logical purposes, such as suggesting energy-efficient and congestion-free routes for data packets;
- Spatial patterns for a 2D basis of images or maps encoded in terms of associated coefficients;
- Patterns to serve as a reference wave to encode data as an interference pattern;
- Patterns of the entire S-Net to be used as a signal carrier or modulator for the computable moving waves.

As pointed out by Neglia et al. [70], the reaction-diffusion model is capable of producing scalable patterns for achieving networked system service goals. These patterns can be generated even when there are failures taking place. They are also highly responsive to changes in the network in terms of quick re-convergence. However, these merits are obtained at the cost of involving continuous communication overhead on updating concentrations of activators and inhibitors.

Since reaction-diffusion algorithms used in different applications share the same mathematical model, their performance cannot be compared in a straight forward fashion. However, researchers can still evaluate their design by examining the mapping from the status and the local observations of nodes to concentrations of activators and inhibitors, and exploring how control parameters of the network are tuned based on dynamics of these concentrations.

The metaphor of pattern formation could be employed by future networked systems that seek a fast and balanced adaptation response to a changing environment through interactions among neighboring components. The components that are first triggered by these changes will spread their influence toward their neighbors, while the neighboring components absorb the influence and deliver their own. The entire system will gradually achieve a new balance, i.e., a new pattern among components that fits the environment and the system goal. The pattern generated from the reaction-diffusion algorithm

that is associated with this metaphor could also be used in signal encoding, carrying, and modulating applications.

### C. Corpse Clustering and Brood Sorting

Corpse clustering refers to the ability of ants to form piles of corpses, or cemeteries. Brood sorting refers to the ability of ants to gather the larvae in a concentric annuli with smaller ones at the center and larger ones at the periphery [76].

There are two founding models for this metaphor. The first one, the Basic Model (BM), was proposed by Deneubourg et al. [77]. In this model, objects are randomly distributed in the area where the ants pick up or drop off according to the similarity of their neighbors. An ant picks up an object with probability  $P_{pu}$  as defined in the following eq. 13.  $k^+$  is the threshold constant for picking up.  $f$  is an estimation function of the fraction of nearby points occupied by objects of the same type. It is in the range from 0 to  $k^+$ . It could be observed from this definition that the probability of picking up an object is increased when this object is mostly surrounded by different types of objects, or when no objects are around at all.

$$P_{pu} = (k^+ / (k^+ + f))^2 \quad (13)$$

$$P_{do} = (f / (k^- + f))^2 \quad (14)$$

An ant drops off a picked up object with probability  $P_{do}$  as defined in the above eq. 14.  $k^-$  is the threshold constant for dropping off.  $f$  is the same function as the ones in the definition of pick up probability with range from 0 to  $k^-$ . It could be observed that the probability of dropping off an object is increased when the ant is located in a point with similar objects. The original model treats the number of the same objects an ant has met in its most recent journey as an approximation for  $f$ . The effectiveness of object clustering was shown in simulations.

The second major model is the Standard Ant Clustering Algorithm (SACA) proposed by Lumer et al. [78], which improves on BM. As defined in the following eq. 15 and 16, the pick up and drop off probabilities are determined according to the similarity of the objects in the ants' immediate regions.  $f(o_i)$  is defined as the average similarity of the object  $o_i$  with other objects in its neighborhood. In addition, a short list of recent drop off points is maintained by each ant. The ants that carry objects tend to move toward the locations where similar objects were dropped.

$$P_{pu}(o_i) = (k^+ / (k^+ + f(o_i)))^2 \quad (15)$$

$$P_{do}(o_i) = \begin{cases} 2f(o_i), & \text{when } f(o_i) < k^- \\ 1, & \text{when } f(o_i) \geq k^- \end{cases} \quad (16)$$

Research efforts have been made to improve the above two *founding models*, focusing on the design choices of (1) the sensing ability of ants [79], [80], (2) the number of objects that could be picked up or dropped off by a single ant [81], [82], (3) the directions the ants travel [83], [84], and (4) the time dependency of the journey of carrying objects [85]. However,

there is no research on comparing the performance of these BIA models.

The nature of corpse cleaning and larval sorting by ants applies to clustering algorithms [86], [87] for data mining and graph-partitioning. Compared to the classic clustering approaches, the corpse clustering and brood sorting inspired clustering algorithms have the advantages of (1) automatically discovering the number of clusters, (2) being robust to the effects of outliers within the data [86], (3) possessing self-organization and decentralization [88], and (4) having no need for prior information. As pointed out by Handl et al. [89], although the runtime of the corpse clustering and brood sorting inspired algorithms scales linearly, their clustering process on low dimensionality data is slower than traditional approaches.

In the intrusion detection system field, Ramos et al. [90] applied the metaphor to the classification of attacks and gained the appealing features of: (1) online and real-time classification, (2) no retraining needed for new attack types, (3) both unsupervised and supervised learning, and (4) a self-organizing nature.

Forestiero et al. [91] built a P2P information system for grid resource organization and discovery. Resource descriptors are disseminated over the grid and spatially sorted according to their keys (i.e., descriptors represented by identical or similar keys are located around neighboring Grid hosts) so that they can be easily found for corresponding resources. The desired distribution on copies of resource descriptors is achieved by agents, which are sent from each peer node and travel randomly in the overlay network, picking (either *copy* or *move*) and dropping resource descriptors among peer nodes. The probability for picking up a descriptor is inversely proportional to the average similarity of this descriptor with those located in the visible region of the current node, while the probability for dropping is proportional. As descriptors are properly reorganized, the operation frequency of agents drops, at the same time, the *move* operation is preferred to the *copy* operation for picking. The system applies a reinforcement learned variable for each agent, called a *pheromone base*. It increases as the operational frequency of this agent (the ratio between the number of operations actually performed and the total number of operation attempted) drops. At the same time, its corresponding threshold for decision making switches from *copy* mode to *move* mode.

The metaphor of corpse clustering and brood sorting could be used in future networked systems that rely on clustering techniques. In addition, it could be used to produce desired data distribution among components in networked systems in order to fit continuously changing environments. This can be achieved by employing agents carrying or dropping data based on similarity with the data in agents' surrounding environment.

#### D. Ant Foraging

Ants are able to find the shortest path to food sites. Dorigo et al. [92] proposed a model called Ant Colony Optimization (ACO), which was inspired by this foraging behavior. The stochastic optimization and problem solving in the model can be reduced to a path finding problem. Basically, ants continuously lay pheromones when they forage. The pheromone

trails evaporate as time elapses. Shorter paths receive more pheromones earlier, since the ants traveling in these trails usually return faster. Ants all probabilistically follow the paths with greater density of pheromones when foraging. The density of pheromones in the shortest path are enhanced, while ones in other paths fade away, because most of the ants choose that trail and continuously lay more pheromones as they travel. At the same time, the enhanced shortest path attracts more ants. This creates a positive feed back loop [93]. This whole path finding process of ants is also a classic example of the *stigmergy* phenomenon [94], in which individuals indirectly communicate with each other through their environment, to which they modify and react. In the original ACO model, the distribution of pheromones over edges in the network influences the *transition probabilities* for ants' selection of the next hop at each node. Ants probabilistically following the transition probabilities adjust the pheromone densities of the chosen edges as they travel and eventually change the distribution of pheromones in the network. The transition probability  $p_{ij}^k(t)$  for the ant  $k$  at node  $i$  selects node  $j$  as its next hop at time  $t$  is defined as follows in eq. 17 [95]:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in \text{allowed}_k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, & \text{if } j \in \text{allowed}_k \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

$\tau_{ij}(t)$  is the pheromone density for the edge  $(i, j)$  at time  $t$ , which represents the reinforcement learned quality of that edge and is updated with the travel time of ants over that edge.  $\eta_{ij}$  is the visibility from node  $i$  to node  $j$ , which represents the length of that edge. The balance between the intensity and the visibility of edge  $(i, j)$  is manipulated by parameters  $\alpha$  and  $\beta$ .  $\text{allowed}_k$  is the tabu list of nodes that have not been visited by ant  $k$ . Since the ACO model targets the classic Traveling Salesman Problem (TSP), the tabu list is applied in the probability calculation instead of the list of node  $i$ 's neighboring nodes. In simulations, the paper demonstrated that with careful control over the number of ants and the decay rate of pheromones, the ACO model could be applied to many difficult problems, such as asymmetric TSP (ATSP), the quadratic assignment problem (QAP), and the job-shop scheduling problem (JSP), to effectively find very promising solutions.

Maniezzo et al. [96] produced a survey paper on ACO as it is applied to these combinatorial optimization problems. In addition, authors investigated the progress on the mathematical verification for the convergence property of ACO. The paper pointed out that formal proofs are not able to fully model behaviors of real complex applications and cannot provide direct guidelines for them either.

The main and most direct application of ACO is routing. ACO-based routing protocols have been applied to circuit-switched networks, such as Ant-Based Control (ABC) [97], packet-switched wired networks, such as AntNet [98], and MANETs, such as AntHocNet [99], HOPNET [100], Ant-Colony-Based Routing Algorithm (ARA) [101] and Termite [102]. All the ACO-based routing protocols inherit the merits from the original metaphor. They are able to perform stochastic route exploration, stochastic data forwarding, and load-

TABLE VI  
POSSIBLE DESIGN CHOICES OF ACO-BASED ROUTING PROTOCOLS

Design Parameters	Choices
(P1) The selection of destinations of forward ants	Random destinations (R) All the nodes in the network (A) Nodes that are close to the source node (C) Local data traffic distribution (T) Requests of session establishment (S)
(P2) The forwarding of the forward ants	Flooding (F) Broadcasting at each hop (B) Following the pheromone gradient (P) Local data traffic distribution (T)
(P3) The forwarding of the backward ants	Retracing the path of forward ant (R) Following the pheromone gradient (P) Flooding (F)
(P4) The executor of pheromone update	Forward ant (F) Backward ant (B) Both (A)
(P5) The direction of pheromone update for forward ants	Source node of the session (S) Destination node of the session (D)
(P6) The direction of pheromone update for backward ants	Source node of the session (S) Destination node of the session (D)
(P7) The strategy of route discovery	Proactive (P) Reactive (R) Hybrid (H)
(P8) The strategy of route maintenance	Proactive (P) Reactive (R) Hybrid (H) Using data packets (D)
(P9) The strategy of link failure processing	Explicit notification (E) Implicit self-healing (I)

TABLE VII  
DESIGN CHOICES MADE BY ACO-BASED ROUTING PROTOCOLS

	P1	P2	P3	P4	P5	P6	P7	P8	P9
ABC	R	P	/	F	S	/	P	P	I
AntNet	T	PT	R	B	/	D	P	P	I
AntHocNet	S	PF	R	B	/	D	R	P	E
Intra HOPNET	C	FP	R	A	S	D	P	P	I
Inter HOPNET	S	PF	R	/	/	/	R	R	E
ARA	S	F	F	A	S	D	R	D	E
Termite	S	PF	P	A	S	D	R	D	I

balancing, and are highly comparable to classic approaches in dynamic networks. The common data structures among these protocols are *pheromone tables*, instead of traditional routing tables. Each pheromone table records the pheromone distribution for one distinguished destination node of the current node as the next hops for data forwarding. In other words, if there are  $m$  possible destination nodes from the current node, there would be  $m$  pheromone tables in this node. If there are  $n$  neighbors of this node, the pheromone table for one destination node  $i$  would have  $n$  entries. The values of pheromone levels in the pheromone tables represent the reinforcement learned quality of possible next hops for the destination nodes. Agents called *ants* are flooded by each node of the network to update these values based on their traveled paths' status. Data packets probabilistically follow the gradient of pheromone distribution at each hop. Although the schemes

are similar among different ACO-based routing protocols, different design choices are made to fit the characteristics of the applied network and the routing objects of design. The differences among ACO-based routing protocols' design, described by the nine design parameters, are listed in table VI and table VII. Besides stochastic route exploration, data forwarding, and load balancing, four main features could be applied to evaluate these different design choices.

**Loop-free Property:** There are four schemes to reduce or prevent loops in the route discovery phase of ACO-based routing protocols:

- When exploring routes, *ant* agents update pheromone tables in the direction of their sources. In other words, supposing that an ant is sent from node  $s$  and comes across node  $i$  and  $j$  in sequence, then the pheromone level for employing node  $i$  as the next hop to node  $s$  from node  $j$  will get relatively enhanced. This scheme was first deployed in ABC. It was adopted by HOPNET, ARA, and Termite. Given enough time, circular routes would mostly be avoided in the network environment with little noise. However, this scheme needs to assume that the link cost in the network is symmetric, and if deployed, would normally make the route from one node to another the same as the one in the opposite direction.
- *Forward ant* agents are sent from source nodes to destination nodes. *Backward ant* agents that travel back to source nodes by retracing the paths of forward ants are used to update the pheromone distribution in the network.

Both the route and the status of a path is recorded in the corresponding forward ant's stack. Intermediate nodes are responsible for cutting route loops in the stack for all forward ants. Thus, when the backward ants update the pheromone tables in the direction of their sources, i.e., the destination nodes of their corresponding sessions, the looping problem is avoided. This scheme was first deployed in AntNet and adopted by the backward ants of AntHocNet. It does not require the assumption of symmetric link cost. In addition, the direction of pheromone updates accords with the original ant foraging behavior. However, stacks in forward ants require extra memory in each node and bring additional transmission overhead to the network.

- HOPNET, ARA, and Termite employ the source update direction scheme. In addition, they choose to deploy another loop prevention scheme where duplicate ants from the same source nodes will be ignored at all intermediate nodes instead of being forwarded. This scheme is effective in avoiding loops in routes, however, it requires extra memory in each node for storing types of visited ants.
- Termite has a unique scheme for dealing with the looping problem. The decision making for the next hop selection tries to avoid cycles. By probabilistically forwarding each data packet following the pheromone gradient toward the destination and repelling the pheromone gradient toward the source, the possibility of packets going into loop routes is greatly reduced. However, this scheme cannot guarantee to produce loop-free routes. The design choices that would affect the loop-free routing feature are P2, P3, P4, P5, and P6, depending on which group of the above four schemes is deployed.

**Traffic Overhead:** The network traffic overhead is a common issue in ACO-based routing protocols, since flooding of ants is the major scheme for routing exploration and maintenance. However, the frequency of sending ants (P7 and P8) and destination nodes of ants (P1) could be tuned in order to reduce this overhead. In addition, the differences in characteristics of the network environment are important for decision making. For example, MANETs differs from packet-switched, wired networks, and circuit-switched networks, in their highly dynamic characteristics. The mobility of the nodes and the fluctuation of the wireless environment makes the topology of the network change often and causes the efficiency of the links to be unstable. To be able to fit these features, the routing protocol should both be adaptive to the dynamic environment and save critical connection and computation resources in the network. This leads to a paradox: On the one hand, frequent updates on the routing information are preferred to get the system prepared for disruptive events. Conversely, less communication is preferred in order to save precious bandwidth and energy, and in order to make the network more scalable. The routing protocols with the former strategy e.g., DSDV [103] and WRP [104], use proactive algorithms, in which routing information is only gathered in response to an event. This type of event includes the request of starting a new session, along with route failures in an existing

session. The routing protocols with the latter strategy e.g., AODV [105], DSR [21], TORA [106], and ABR [107], use reactive algorithms, in which routing information is collected periodically. There are routing protocols such as ZRP [108] and SHARP [109], which implement both strategies in a hybrid manner in order to take advantage of the merits of both. Like the classic approaches, ACO-based routing protocols make decisions on the sending frequency and destinations of ant agents based on the characteristics of the network to which they apply: For establishing or maintaining routes, ABC sends ant agents continuously for all nodes in the network. AntNet continuously sends ant agents for the destination nodes based on the local traffic distribution of each node. For MANETs, the hybrid protocol AntHocNet chooses to send ant agents for both route exploration for destination nodes under requests of session establishment in *reactive path setup* phase. In addition, the protocol sends ant agents for route verification of paths that are obtained by continuously exchanged *hello messages* among neighboring nodes in the *proactive path maintenance and exploration* phase. HOPNET divides the entire network into zones. It executes proactive intra-zone route updates and reactive inter-zone route updates by employing ant agents. A node may be within multiple overlapping zones and zones could vary in size, as determined by the radius, i.e., the number of hops from the center node in the zone. The keys to balancing proactive and reactive routing, along with reducing the routing overhead in HOPNET are the radius of the zone and the degree of overlap between zones. The paper indicates that when they are properly tuned, HOPNET can achieve scalable and effective routing at the same time. Along with these hybrid routing protocols for MANETs, the light-weight on-demand routing protocols, ARA and Termite, use data packets instead of ant agents, to perform route maintenance, in order to reduce the extra cost on maintenance traffic.

**Memory Consumption and Scalability:** AntNet and AntHocNet require extra memory for the stack of each *forward ant* agent. In addition, AntHocNet exchanges the memory of *virtual pheromone tables* in order to continuously improve existing paths and explore new paths for the multiple-path stochastic data forwarding of each session. The virtual pheromone tables are updated by using *hello messages* among neighboring nodes of the network. Routes in the virtual pheromone tables are verified by ants before being put into the pheromone table for data forwarding. ACO-based routing protocols for circuit-switched (ABC) or wired packet-switched (AntNet) networks tend to use *static routing tables*. In this scheme, each node contains pheromone tables for all the other nodes in the network. As a result, this static pheromone table scheme does not scale well. ACO-based routing protocols for MANETs (AntHocNet, HOPNET, ARA and Termite) tend to use *dynamic routing tables*. In this scheme, each node only maintains pheromone tables for the nodes whose routes are explored by its ants.

Even though ACO-based routing protocols have been proven to be compatible with classic approaches, most evaluations of the ACO-based routing protocols are performed through simulation, with assumptions on either the network environment or other protocol layers. In addition, there is no universal evaluation through experiments for the performance

comparison amongst all these different ACO-based routing protocols and classic approaches.

Along with ACO-based approaches for underlay network, Michlmayr [110] applied the AntNet scheme to the query routing task in P2P networks. Basically, each node in the network maintains the metadata that annotates the resources present at that node. Queries to find the locations of resources are sent and forwarded at each intermediate node to peers that are likely to store corresponding resources as *forward ant* agents. The next hops are determined based on the static pheromone tables at each node, which are in the size of  $C \times n$ , with  $C$  possible resource-node pairs in the network and  $n$  neighboring peers. *Backward ant* agents retrace corresponding forward ants to the source nodes to answer the resource location queries. In addition, they update the pheromone distribution of nodes along their journey with the goodness of the paths (the number of resources and the lengths of the paths) for further query routing. In simulations, the paper showed that the ratio of exploring to exploiting forward ants does not statistically influence the efficiency of the entire P2P query routing protocol.

The metaphor of ant foraging is well modeled by ACO algorithms, which could be directly applied to the problem of routing in future networked systems. Future research efforts could be placed on providing loop-free property, reducing traffic overhead and memory consumption, and improving scalability of the ACO-based routing schemes. Evaluating the performance of ACO-based routing scheme through experiments on actual devices also offers space for future research.

#### E. Bee Foraging

Similar to the foraging of ants, honey bees are able to find close follower sites. This foraging behavior has inspired data forwarding mechanisms in networked systems as well. BeeHive [111] is a hierarchical routing protocol for packet switched networks inspired by the foraging behavior of bee colonies. Basically, there are two types of agents for route discovery and update, *short distance bee agents* and *long distance bee agents*, which are identified by their time to live (TTL). The entire network is divided into non-overlapping clusters of nodes called *foraging zones*, in which nodes could reach each other through *short distance bee agents*. Each *foraging zone* chooses the node with the lowest IP address as the representative node. Representative nodes could reach each other through *long distance bee agents*. In each *foraging zone*, non-representative nodes continuously send out *short distance bee agents* to explore and maintain routes to other nodes in the same cluster. The representative node keeps on sending out *long distance bee agents* to explore and maintain routes to other representative nodes from different clusters. Bee agents are re-broadcasted at each hop. They update routing tables in the direction of source nodes. Duplicates of bee agents are killed at the destination nodes. Data packets are forwarded probabilistically at each hop following the nodes' routing tables. In simulations, BeeHive achieved better performance with less memory consumptions than AntNet in dynamically changing networks.

Another major application of bee-inspired algorithms employs the metaphor of the bee dance. When a honey bee returns

to its nest after encountering a food site, it will perform a *waggle dance* for that site in order to attract more honey bees to forage toward it. The dance contains information about the distance, the direction and the quality of the site. It will be performed by the honey bee only when the site quality exceeds a certain threshold [112]. BeeAdHoc [113] is a reactive routing protocol for MANETs in which a dance is abstracted as the number of clones that could be made for agents. There are two main types of agents in this system: foragers and scouts. Foragers are responsible for carrying data back and forth between nodes, following the routes provided by the scouts or other foragers. When a forager returns after its journey, it recruits new foragers with the same route and type by dancing. The number of the agents cloned is determined according to the quality of the path the forager traversed. As a result, the route with the higher quality will have more foragers to transfer data between the source and the destination. One clone of the selected forager will be consumed for the data forwarding and the dance number of the forager is reduced by 1. If the dance number is 0, the original forager is selected and there will be no forager of this route for further data forwarding. This scheme ensures that young foragers, who represent the latest routes and are likely to remain valid in the future, are favored over older ones. If there is no forager that knows a route for carrying data to a destination node, the current node will flood scouts through the network to discover routes to the destination node. At each intermediate node, the scouts are re-broadcasted and each scout has the TTL parameter, which indicates the number of times it could be re-broadcasted. The destination node sends back all of the scouts it received to ensure the discovery of multiple paths. When a scout gets back to the source node, it will recruit a forager and hand the route over.

BeeAdHoc was evaluated analytically, through simulation, and experimentally in [114] with state-of-the-art routing protocols for MANETs. Although the computational complexity analysis has not been performed, the formal verification model in the paper proved that the distribution of foragers, which travel through different routes for the same source and destination pairs, reflects the quality of routes. In addition, the paper formally proved that the data forwarding pattern could be optimized in this model. In the simulation, it was proven that BeeAdHoc requires significantly less energy than DSR, AODV, and DSDV, while achieving indistinctive differences in performance metrics such as packet delivery ratio, delay and throughput. In the experiment, BeeAdHoc was implemented in the Linux network stack. It could consistently perform routing comparable to OLSR and AODV, while consuming less energy and employing less control packets. Readers interested in this area could consult the Farooq's 2008 book [115], which summarized the applications applying bee-inspired algorithms for computer networking.

Recent research in biology [116], [117], [118] showed that the traditional hypothesis of bee foraging behavior, that foragers would use the flower sites' positional information from other foragers for food hunting, is not correct. Instead, foragers might solely rely on their own information of food sources obtained from their previous experience. The positional information of flower sites from other peers is only

used as a backup. Analytically, the latter foraging model might be more efficient than the former one in nectar-poor environments, since it produces a relatively sparse distribution of foraging flower sites and prevents over harvesting of the resources. However, in the environments with rich quantities of nectar per flower, the benefits of the latter model might not stand.

Bailis et al. [119] used simulation to compare these two models: (1) the *share position* model, in which foragers become active at the dance advertised position, and (2) the *private position* model, in which foragers ignore the dance's positional information on the flower sites and forage toward previously known positions. If there are no known sites, foragers will play the role of scouts. The paper gives four suggestions on important factors for choosing the appropriate model based on the simulation results, which could be applied to different networking environments when bee dance inspired algorithms are applied to networking problems:

- When the concentration of nectar in the flower sites increases, the importance of foraging from different food sources decreases, since the depletion slows, while the benefit of sharing high-quality food sites increases.
- When the flower sites could be easily found, even though the depletion is high, oversubscribing could still do less harm since replacement food sources could be explored.
- In a heterogeneous environment, where the difference between the best and the worst sites is very large, pursuing high-quality flower sites may outweigh the cost of over-harvesting. While in a homogeneous environment, the effect is not that obvious.
- In a clustered environment, where flower sites are located closely together in each cluster and sparsely among different clusters, the importance of foraging from different food sources decreases compared to a scattered environment, since it is easier to find replacement sites nearby. However, it might be difficult to discover another flower cluster foraging source after a cluster is depleted.

Jevtic et al. [120] applied the above metaphor to the problem of distributed task allocation in a swarm of robots to find targets in a 2-D arena. Both the distance between a robot and targets in the field, along with qualities of these targets are used to determine probabilities of selecting available targets. There are two parameters in the calculation to bias the decision making toward the distance cost or the quality reward. These parameters are determined based on the initial distribution of robots and their targets in the area, which is in the same way as how Bailis et al. [119] select appropriate bee dance models to fit scenarios with different flower distributions. The authors proved that this algorithm is scalable in terms of the number of robots and their targets, along with being adaptive to the quality of targets, in both simulations and experiments on actual robots.

The metaphor of bee foraging is well modeled by the scouts and foragers model, which could be directly applied to the problem of routing in future networked systems. Like ACO-based routing protocols, the future research efforts for improving bee foraging inspired routing schemes could also be placed on providing loop-free mechanisms, reducing traffic overhead and memory consumption, and improving scalability.

TABLE VIII  
PERFORMANCE COMPARISON AMONGST FORAGING BIAs

	Traffic Overhead	Memory Consumption	Scalability
ABC	High	Medium	Low
AntNet	High	High	Low
AntHocNet	Medium	High	Medium
HOPNET	Medium	Low	High
ARA	Low	Low	Medium
Termite	Low	Low	Medium
BeeHive	High	Medium	High
BeeAdHoc	Low	Low	Medium

In addition, recent research on how foragers select their routes for data forwarding provides a new angle for designing bee-inspired routing schemes that fit the data forwarding requirements in future networked systems. Table VIII shows the performance comparison amongst different foraging inspired algorithms.

Farooq et al. [121] produced a comprehensive survey paper on foraging inspired routing algorithms for both wired networks and wireless MANETs, and comparison to classic data forwarding mechanisms. As the paper pointed out, foraging inspired algorithms have advantages in failure resilience, efficient resource utilization, higher throughput, and adaptiveness toward changing traffic. However, it is hard to formally verify emergent properties of these BIAs. In addition, there are few experiments on physical implementation of them, which raises concerns on their performance in real environments.

In simulations, Wang et al. [122], [123] evaluated the performance of foraging inspired routing algorithms for MANETs. In the perspective of energy consumption, the authors drew the conclusion that algorithms that select next hops from neighboring peers within the free-space-fading distance threshold are able to save energy at the cost of a few more hops for data forwarding.

In comparisons among foraging inspired algorithms, such as BeeAdHoc, AntAdHoc, and ARA, along with classic data forwarding schemes, such as AODV and DSR, similar performance is achieved in terms of packet delivery ratio, goodput, and throughput in MANETs with constant bit rate UDP traffic. Nevertheless, foraging inspired algorithms outperform classic approaches in terms of end-to-end delay, routing overhead, delay jitter, and scalability.

The authors also proved that among different foraging inspired algorithms, BeeAdHoc achieves the lowest end-to-end delay due to its simple data structure and less use of control packets. Both ARA and BeeAdHoc have lower routing overhead compared to other BIAs in simulation. This is because the number of control messages needed for ARA decreases as the mobility of nodes decreases. In addition, foragers are sent back only when destination nodes need to communicate with source nodes, leading to a large decrease of routing overhead. AntHocNet obtains lower delay jitter than other approaches even when there is a large number of participating nodes because of its proactive route maintenance procedure.

#### F. Pulse-Coupled Oscillator

There are many biological systems showing synchronized behaviors [124], such as the synchronization of blinking

among fireflies [125], a synchronous firing of neurons [126], [127], and the synchronization of heart cells beating [128].

Researchers are adapting the rules from these natural systems to synchronicity in the network, which is a useful primitive for communication scheduling, coordinated duty cycling, and time synchronization. Basically, distributed synchronization is commonly featured by the pulse-coupled oscillator (PCO) model. Take PCO in fireflies for example, where an oscillator represents an internal clock which indicates when the firefly should flash. Upon reception of a pulse only from neighboring oscillators, this clock is adjusted. Over time, synchronization emerges under the appropriate clock adjustment rule.

The theoretical PCO model represented by Mirolo and Strogatz [129] explains how individuals such as neurons and fireflies spontaneously synchronize. In PCO, all the nodes act as oscillators with a common fixed time period  $T$ . The oscillator has an internal time phase  $t$ , which starts at zero and increments at a constant rate until  $t = T$ , when the node fires and resets  $t$  back to zero. Nodes may start at different times and the goal of the system is to synchronize the phase  $t$  among nodes. If a node detected a firing from its neighbor,  $t$  in this node is adjusted forward. In other words, the node's time to fire is shortened. The new value of the time phase  $t_{update}$  is set based on the following eq. 18:

$$t_{update} = f^{-1}(f(t) + \epsilon) \quad (18)$$

$f$  is called the *firing function*, which should be smooth, monotonically increasing and concave downward.  $\epsilon$  is a small positive constant which should be less than 1. If the new phase value is bigger than  $T$ , then the node fires immediately and resets  $t$  back to zero. In mathematical terms, the PCO model guarantees convergence [129], however, it could not be appropriately applied to synchronization of ad hoc networks, until the following realistic issues were settled.

**Topology:** Mirolo and Strogatz [129] claimed that in order to make any number of nodes converge to the same phase, the network needs to have full mesh topology, i.e., each node could observe all others' firing, which is unrealistic in many ad hoc networks. Lucarelli and Wang [130] demonstrated that the all-to-all communication constraint [131] could be relaxed in multi-hop topologies, i.e., each node only needs to observe firings from its neighbors.

**Delays:** There are transmission delays  $T_{Tx}$ , decoding delays  $T_{dec}$ , and propagation delays  $T_{prop}$  between a sending node firing and a receiving node identifying the transmitted synchronization word. The last type of delay can be neglected in comparison to the former two. In the theoretical PCO model, even propagation delays among nodes could lead to an unstable system status where nodes continuously fire one after another [132], [133]. To resolve this issue, Werner-Allen et al. [134] applied timestamps on the MAC layer that record the MAC delays experienced by messages prior to transmission. These timestamps are used by receiving nodes to estimate the firing time of their corresponding sending nodes. This scheme could make accurate decisions on  $T_{Tx}$ , however,  $T_{dec}$  is not counted. Tyrrell et al. [135] estimated the total delay  $T_{del}$  as the sum of the  $T_{Tx}$  and  $T_{dec}$  whose values are set identically

among nodes. After firing, a node delays its transmission of the synchronization word. The waiting delay ( $T_{wait}$ ) is chosen to be  $T - T_{del}$  in order to make sure that the receiving nodes recognize their neighbors' firing approximately  $T$  seconds after a transmitting node fires. This scheme assumes all the delays among nodes are identical, which is not realistic in many ad hoc networks.

For future research, a better understanding of either the scheme of recording the entire delay time, or the procedure of producing statistical estimates on the delays, needs to be established. Because of the delays, the nodes cannot react instantaneously to the received firing. A node might have already fired and thus no longer be able to react to the delayed firing. If the received firing is out of order, a node needs to undo and redo the adjustments in order to perform a correct time phase advance, which quickly makes the algorithm complicated and unmanageable. Werner-Allen et al. [134] solved this problem by proposing a reachback response scheme: When a node hears a neighbor's firing event, it places the message in a queue, timestamped with the correct internal time at which the firing event of the neighbor occurred. When the node reaches time  $t = T$ , it fires, waits for the grace period, computes how much it would have advanced its phase based on the messages stored in the queue in one shot, and performs the increase of its time phase as a PCO model does.

**Simultaneous Firing:** Tyrrell et al. [135] claimed that the best case for the theoretical PCO model, that all nodes fire simultaneously, is actually the worst case for channel contention, since it creates the potential for many collisions, resulting in large message delays. Thus, a random transmission delay to node firing messages at the application level is added in order to avoid repeated collisions and to control the extent of message delay. This scheme introduces an extra value on  $T_{Tx}$ . On the contrary, Tyrrell et al. [135] believed that the best case for PCO is affected by collisions in a similar way to flooding. Indeed, from the physical layer point of view, receiving nodes far away would benefit from the sum of transmitted powers produced by all nodes transmitting a common synchronization word simultaneously [136]. However, due to the fact that a node could not receive while transmitting, the best case for PCO still needs to be avoided. This avoidance is implemented by setting the time period for firing of each node to be  $2 \cdot T$ , where  $T$  denotes the synchronization period. It also splits oscillators into two groups whose time difference is  $T$ . In other words, each group will fire  $T$  seconds apart, facilitating synchronization. In order to achieve this equilibrium, when a message is successfully received during the listening time of a node, the time phase  $t$  of its inner oscillator abruptly shifts to a new value  $t_{update}$  as in the following eq. 19.

$$t_{update} = (t + T) \% (2 \cdot T) \quad (19)$$

Table IX summarizes the performance comparison amongst PCO BIAs. The above described models are grouped into three different types: (1) theoretical PCO models [129], [130], [131], (2) the time-stamp PCO models [134], and (3) the estimation PCO models [135].

The PCO model has been directly applied to the synchronization for cycle-based execution of distributed protocols

TABLE IX  
PERFORMANCE COMPARISON AMONGST PULSE-COUPLED OSCILLATOR BIAs

	Theoretical Model	Time-stamp Model	Estimation Model
Consideration on $T_{TX}$	None	Accurate time-stamp	Identical estimation
Consideration on $T_{dec}$	None	None	Identical estimation
Out-of-order Firing Handling	None	Store firings and advance time phase in one shot	Frequent adjustments upon receiving firings
Collision Avoidance	None	None	Use random transmission delay

in P2P overlay networks [137], along with periodic data gathering in WSNs [138]. The feature of converging time phases among distributed nodes through local observation was also employed to obtain consensus [139]. In this sensor network application, each node encodes its local decision of detection into its time phase. With the PCO model, the entire network would eventually converge to an equilibrium where the phase differences among nodes become zero. The common value of time phases is mapped to a consensus in decision making process.

Yu et al. [140] employed the methodology of obtaining consensus through a high degree of local observations and communications to the problem of *spectrum sensing*. Without the prior knowledge of primary users nor a centralized control, each secondary user takes individual measures in a target spectrum band and communicates with neighbors iteratively with observed power values. Eventually, they are able to obtain an average consensus in order to determine the existence of primary users in the band. The authors proved that the algorithm can achieve an exponential convergence rate. In addition, the information from the neighbor, whose measurement has the maximum deviation from the mean value of all of its neighboring peers, is ignored. This scheme makes the system resistant to various Spectrum Sensing Data Falsification (SSDF) attacks. Moreover, the design has been proven to be robust toward link failures in the collaborative sensing network [141].

Degesys et al. [142] proposed a decentralized algorithm based on the PCO framework for time and event desynchronization. Instead of attempting to converge on the same time phase, oscillators try to fire as far away as possible from each other. To achieve this equilibrium, each oscillator listens to the firing before and after it fires and adjusts its time phase to be the midpoint of the firing before and after its own. The algorithm is applied to TDMA scheduling. The authors showed that, compared with other TDMA protocols, the DESYNC TDMA is fair, collision-free, and supplies high throughput under heavy loads, predictable latency, and self-adjusting protocol without explicit scheduling or time synchronization. In addition, it outperforms in energy efficiency by utilizing asynchronous low power listening.

The Reference Broadcast Synchronization (RBS) model [143] is the non-biologically-inspired, classic, centralized approach corresponding to PCO. Theoretically, RBS achieves

a high accuracy in synchronization without being affected by access delay as PCO is. However, for large scale networks, nodes in multi-hop RBS are required to be in the communication range of base nodes in their own clusters. These nodes must also be able to receive time difference messages from base nodes of neighboring clusters relayed by the corresponding gateway nodes. The PCO model only relies on local observation. Mutazono et al. [144] compared these two models in simulations and found that the synchronicity of Multi-hop RBS is easier to be obtained in high node density environments. Nevertheless, PCO is stable among different node distributions. In addition, although both models are robust to frequent packet loss, PCO is more effective in an unstable communication environment where tight time synchronization is required.

Breza et al. [145] demonstrated that emergent algorithms can be combined to control multiple system parameters in an integrative manner. The authors nested gossip algorithm [146] into the PCO model and applied it to time synchronization for the Sun Small Programmable Object Technology (SPOT) sensor network. The paper showed that the model's performance is not as good as evaluated in simulations in terms of converging time and stability of synchrony. Assumptions of the BIA fail because of the clock drift and impacts from other high priority processes in SPOT nodes. Indeed, many applications of BIAs in real world scenarios suffer from similar problems. Researchers that are interested in employing these BIAs should verify that properties in the target networked system do not break requirements for keeping algorithms functioning.

The metaphor of pulse-coupled oscillators could be employed by future networked systems that target time synchronization or consensus building among components. The PCO model that is associated with this metaphor is able to converge time phases among distributed components through local observation and adjustment. The consensus obtaining mechanism is able to achieve consensus in the system through local synchronizing interactions among components, based on the encoding of each component's local decision into its time phase.

#### G. Information Epidemics

Epidemic is the process that an infectious disease spreads among a population. Researchers are working to build epi-

demic models in order to estimate the spread of information in computer networks. Different models were built to capture the infection pattern of diseases [147]. Basically, for each infection type, the entire population is divided into five different epidemiological classes: (1)  $M$ : individuals who are born with temporary passive immunity to the infection; (2)  $S$ : individuals who are susceptible to the infection. This class consists of the individuals who were not born with passive immunity, along with the individuals from  $M$ , whose maternal antibodies disappear from their bodies over time; (3)  $E$ : exposed individuals from  $S$  who are infected but not yet infectious; (4)  $I$ : infectious individuals from  $E$  who are capable of transmitting infection to other individuals in  $S$ ; and (5)  $R$ : individuals who are permanently immune after experiencing the infection. For certain infection types, individuals in  $E$  and  $I$  are able to recover and gain permanent immunity, which means they could move into  $R$ . For other types of infections, individuals in  $E$  and  $I$  might still be susceptible after they recover, which means they could move into  $S$ . To make modeling of the susceptible-infective interaction focused, both class  $M$  and  $E$  are often omitted. Based on the flow patterns between compartments, the most discussed models are *SIR*, *SIRS*, *SI*, and *SIS*. Those models have been well analyzed under different assumptions, such as age, heterogeneity, and spatial structures.

There are two major applications of information epidemics: (1) modeling the spread of computer viruses [148], [149], [150], [151], [152], and (2) routing in highly dynamic MANETs, i.e., opportunistic networks [153]. The latter have recently received considerable attention. Indeed, epidemic routing is an effective context-oblivious scheme for data forwarding in disconnected or intermittently connected networks. In these scenarios, nodes are mobile and links are highly unstable. The networks are so dynamic that the data forwarding between a source node and a destination node needs to be built without the guarantee of the existence of a corresponding simultaneous path between them. Moreover, because of this dynamic nature, nodes that are often isolated could hardly get timely information about their context in the network, e.g., topology information. These characteristics make the design of routing schemes more focused on reliability and robustness than on performance and resource consumption.

Epidemic routing algorithms implement end-to-end data forwarding, under (1) no guarantee of the existence of end-to-end routes, and (2) no guarantee of the existence of network topology information, by mimicking the spread pattern of disease among indirectly contacted individuals. Basically, nodes in the network could be considered individuals that constantly move and contact different nodes. The communication between two temporally linked nodes could be considered the infection process, during which replicates of data packets from one node are forwarded and stored to another node. Eventually, more and more nodes that carry the redundant data from the source node could move to the location within the communication range of the destination node. The data forwarding can be accomplished even if there were no end-to-end routes from the source node to the destination node initially.

The work of Vahdat et al. drew research attention to the functionality and efficiency of epidemic routing [154]. In their

design, each node maintains a buffer for storing the messages to be transmitted. Each message is identified by a 32 bit unique number and possesses a hop count number like the TTL in the IP protocol, which represents the maximum number of forwarding hops a message may travel. By employing the message identifier, a hash table in each node is maintained to index the buffered messages. In addition, a summary vector is generated to compactly represent all the messages in the buffer. To avoid redundant connections, each node also maintains a list of hosts that it has contacted recently. When two recently unseen nodes meet, a three-way *Anti-Entropy Session* takes place. One node  $A$  transmits its summary vector to the other node  $B$ . Node  $B$  calculates the set difference between its own summary vector and the received summary vector, and requests messages that it does not have from  $A$ . Node  $A$  then transmits to  $B$  the requested messages. Their hop count number is reduced by 1. Finally, node  $B$  transmits its messages that node  $A$  does not have to  $A$ . The hop count number of these messages is also reduced by 1. This anti-entropy session would happen again when node  $A$  and  $B$  meet new nodes. These schemes attempt to maximize the percentage of delivery. The goal of minimizing resource consumption is maintained by only allowing anti-entropy sessions to take place between recently unseen nodes and controlling the hop count number. In simulations, the authors demonstrated that under the preset topology and networking dynamics, the protocol could achieve high end-to-end delivery rates. Additionally, the simulation showed that high hop count would result in congestion, while low hop count would result in increased latency.

Effort has been invested in improving epidemic routing schemes for less resource consumption with less sacrifice on the delivery rates guarantee. Two directions have been attempted: (1) reducing the number of hops for data forwarding before the messages reach the destinations; and (2) reducing the amount of data needed to be transmitted for the spread of information.

For the first direction, two-hop routing is the primary scheme in which messages are only allowed to make two hops from the source nodes to the destination nodes. Grossglauser et al. [155] demonstrated that the two-hop epidemic routing scheme could work well in large scale opportunistic networks. Groenevelt et al. [156] modified the original design to allow the source nodes to forward a copy of messages to all nodes they encounter instead of to only one node they encounter. These nodes would perform the forwarding again when they meet the destination nodes. The authors managed to build a stochastic model that demonstrates the relationship between the message delay, along with the distribution of the number of copies of the message at the time the message is delivered. This two-hop epidemic routing scheme has been evaluated under the assumptions on the lifetime of messages [157] and the distribution of nodes' inter-meeting times [158]. Spyropoulos et al. [159] designed a new scheme on top of the existing multiple two-hop epidemic routing protocol. It controls both the number of copies of messages from source nodes, along with the amount of copies that must be forwarded on each encounter. The authors proved that by appropriately manipulating these two primary parameters, a balance between delivery rates and resource consumption can be achieved.

TABLE X  
PERFORMANCE COMPARISON AMONGST EPIDEMIC ROUTING BIAs

	IMMUNE	VACCINE
Spread of Antipackets	Local	Global
Storage Requirements	High	Low
Power Efficiency	High	Low
Optimal Scheme to Improve Performance	Time-limited forwarding	Probabilistic forwarding

For the second direction, coding schemes have been applied to reduce the amount of data that needs to be forwarded for the successful delivery of information. *Erasure coding* [160] was first employed in epidemic routing by Wang et al. [161]. It produces a number of data blocks for the original message so that their entire size is smaller than the one of the original message. As a result, it is possible to retrieve the original message even if only a subset of blocks are obtained. Jain et al. [162] proved that by applying the erasure coding to the epidemic routing scheme, the delivery rate could be improved when transmission failures take place.

Network coding [163] is another coding scheme that has been applied to epidemic routing, in which nodes send out linear combinations of packets they receive. A simple example would be a communication procedure among three nodes *A*, *B*, and *C*. Suppose node *A* is sending message *a* out while node *C* is sending message *c* out. After node *B* receives both messages, it forwards one message as *a xor c* instead of two original ones. Both node *A* and *C* could retrieve their expected messages while the number of transmissions is reduced. Widmer et al. [164] proved that the network coding based epidemic routing could reduce resource consumption.

In addition, Haas et al. [165] proposed a Shared Wireless Infostation Model (SWIM) for data acquisition in wireless ad hoc sensor networks. The paper showed that through the deletion of obsolete information on participating nodes, the storage cost of carrying packet copies for reducing the packet delivery delay to data sinks can be reduced to a moderate level.

In the network, packets act as infectious diseases among nodes. A node is *infected*, i.e., in epidemiological class *I*, if it carries a copy for a packet. A node is in the class *S* if it can carry packets for other nodes. A node is *recovered*, i.e., in class *R*, if it has delivered a copy for a packet to a data sink. In principle, a packet cannot be stored twice on the same node. In other words, nodes become immune to copies of packets that they received before. After determining the contact rate between nodes and the contact rate between nodes and data sinks, researchers are able to determine the time needed for the successful offloading of certain packets at a desired confidence level. Using this information, storage requirements of participating nodes can be determined for the system.

By employing Markov chains for the performance comparison, the paper identified feasible deletion schemes for obsolete information in the SWIM model. The *IMMUNE\_TX* scheme achieves high power efficiency, in which nodes retain identifiers called *antipackets* for packets that they offloaded in order to refuse receiving further copies i.e., nodes are in *IMMUNE*

mode for obsolete information deletion. In addition, they share these identifiers with neighbors who carry same copies. The *VACCINE* scheme greatly reduces storage requirements in the storage-delay trade-off, in which nodes share their identifiers with all of the peers within transmission range.

Zhang et al. [166] employed ordinary differential equations to further evaluate the performance of epidemic routing under different forwarding and recovery schemes in terms of their impact on the trade-off between delivery delay and resource consumption. In addition, the paper verified that with the appropriate choice on buffer management schemes, the effect of small buffer space in participating nodes on delivery performance can be reduced to a negligible level.

In scenarios where the *IMMUNE* scheme is used for obsolete information deletion, employing the global timeout extension for carrying packets and their corresponding identifiers can effectively control the number of packet copies at the cost of increased delivery delay. The limited time forwarding extension for locally determining the period of carrying packets is useful for reducing the consumption on buffer space of participating nodes with a small increase in the delivery delay. In other words, when the system is constrained by buffer occupancy, time-limited forwarding should be selected to improve its performance.

The *VACCINE* obsolete information deletion scheme requires less buffer occupancy and reduces the number of packet copies compared to the *IMMUNE* mode. The resource consumption difference is large when the probabilistic forwarding extension is used in the epidemic routing, in which neighboring nodes carry packet copies for each other with identical probability *p*. In other words, when the system is constrained by power consumption, probabilistic forwarding should be used for boosting its performance. Table X summarizes the performance comparison amongst different epidemic routing algorithms.

The metaphor of information epidemics could be used for routing in future opportunistic networks and understanding the spread of new viruses or information in computer networks. Future research directions for this metaphor could be to improve the delivery rate of opportunistic routing while consuming less resources, along with building models for demonstrating the spreading patterns of viruses or information in future networked systems.

#### H. Summary

The performance differences amongst division of labor, foraging, pulse-coupled oscillator, and epidemic inspired algorithms are compared in tables V, VIII, IX, and X. There is minimal literature comparing the performance of algorithms inspired by the pattern formation metaphor. We argue that this might be caused by the fact that applications that employs the reaction-diffusion algorithm share the same mathematical model. There is little literature providing an universal comparison amongst different corpse clustering and brood sorting algorithms through simulations and experiments. We suggest that this field has research room for future work. Advantages to classic approaches, limitations, and border conditions of the above two BIAs, along with other algorithms inspired by homogeneous biological organisms, are evaluated in table III.

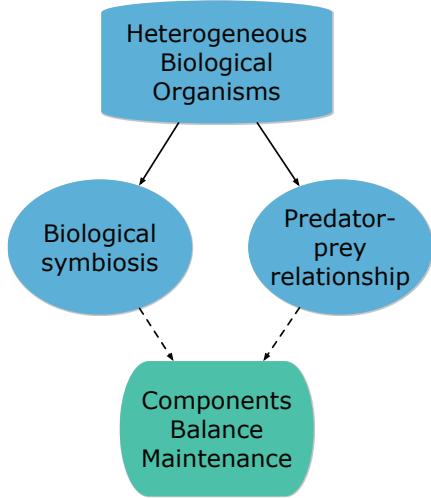


Fig. 5. A Taxonomy of Algorithms Inspired by Heterogeneous Biological Organisms (blue) with Their Applications (green)

#### IV. ALGORITHMS INSPIRED BY HETEROGENEOUS BIOLOGICAL ORGANISMS

In this section, we review BIAs inspired by the interactions among distributed heterogeneous individuals from different colonies, through which the dynamic balance amongst different colonies are maintained. The patterns and relationships among individuals in these biological colonies could be used to understand computer networking better and to guide its development. The taxonomy of these algorithms is provided in figure 5.

Usually, when applying these BIAs to the computer networking field, mapping the components of the networked systems to the individuals of the biological colonies is not necessarily straight forward. It requires creative designs. Table XI summarizes networking applications that are suitable to employ algorithms inspired by heterogeneous biological organisms.

##### A. Predator-prey Models

The population relationship between predators and their prey is described by the Lotka-Volterra (LV) competition model [167], [168]. In the scenario of  $n$  different types of predators, one predator species' fitness is affected by the presence of other predator species that compete for at least one kind of limited prey, modeled as follows in eq. 20:  $x_i(t)$  denotes the population density of the  $i$ th species.  $r_i(t) = b_i(t) - d_i(t)$  is the intrinsic growth rate. It is stable in the absence of all other competing species.  $b_i(t)$  and  $d_i(t)$  are the respective birth and death rates.  $a_{ii}(t)$  is the rate of intra-specific competition of species  $i$ . Each species reproduces itself proportionally to its own population in this intra-specific competition.  $a_{ij}(t)$  ( $i \neq j$ ) is the rate of inter-specific competition between species  $i$  and  $j$ . Species  $i$  reproduces proportionally to the population of species  $j$  in this inter-specific competition [169].

$$\frac{dx_i}{dt} = x_i(t)[r_i(t) - \sum_{j=1}^n a_{ij}(t)x_j(t)], \quad i = 1, 2, \dots, n \quad (20)$$

This metaphor is mostly used for congestion control in computer networks.

Antoniou et al. [170] proposed a congestion prevention scheme for WSNs by regulating the rates of traffic flows hop-by-hop according to the deterministic LV competition model. In this approach, nodes in the network take different roles as Source Nodes (SNs), Relay Nodes (RNs), and Source-Relay Nodes (SRNs). SNs and SRNs generate data packets, which are relayed to the sink at either non-packet generating RNs or SRNs. Each node is in charge of self-regulating and self-adapting the rate of its traffic flow, i.e., the rate at which it generates or forwards packets. The traffic flows from different nodes compete for available buffer capacity at one-hop-away receiving *parent nodes*, which are involved in the path leading to the sink.

For the metaphor mapping, the entire WSN is considered an ecosystem and is divided into a number of smaller sub-ecosystems. Each sub-ecosystem consists of all nodes that send data traffic to a particular parent node. The traffic flows initiated by each node in the sub-ecosystem are considered competing species. The buffer capacity of the parent node is considered limited prey. The packet sending rate is considered the population density of the species, i.e., its traffic flow, and is affected by interactions among competing flows and the available buffer capacity. The possible states of the applied LV model are (1) all species in the system survive and coexist; or (2) at least one species survives, out-competes the rest, and condemns them to extinction [171]. The proposed approach pursues the surviving and coexisting state of all species in the system. It has been proven that when the rate of intra-specific competition is larger than the rate of inter-specific competition, the  $n$ -species LV system's traffic flow rates generated by each SN will converge to a global and asymptotically stable coexistence solution [172]. The convergence time is found to be proportional to the rate of inter-specific competition and to be inversely proportional to the intrinsic growth rate of each species. Traffic flow rates of SRNs and RNs are determined in a similar way as SNs. It was also proven in simulations that this mechanism is able to achieve high packet delivery ratios, low end-to-end delay, scalability, and fairness among competing flows. Furthermore, the proposed scheme was found to outperform the classic Additive Increase Multiplicative Decrease (AIMD)-like, rate-based congestion control approaches [173], [174], [175]. These approaches provoke a QoS violated, long time converging, saw-tooth rate behavior, for WSNs in terms of stability and flow rate smoothness. As demonstrated in the paper, the stability of the system equilibrium highly depends on relationships amongst competition rates of species. As a result, in order to make future networked systems that employ the predator-prey inspired algorithm converge fast, and be highly responsive and robust against changes, competition rates in the model must be well chosen.

##### B. Biological Symbiosis Models

Biological symbiosis could be represented by the phenomena of bacteria coexisting in a shared medium. Yomo et al. [176] provided a mathematical model for describing the

TABLE XI  
SUITABLE NETWORKING APPLICATIONS OF ALGORITHMS INSPIRED FROM  
HETEROGENEOUS BIOLOGICAL ORGANISMS

Biological Schemes	Application Fields
Predator-prey relationship (LV competition model)	Dynamic balance maintenance among heterogeneous components that compete for limited resources
Biological symbiosis model	Dynamic balance maintenance among heterogeneous components that rely on each other

interaction between two types of bacteria that coexist by exchanging metabolites through a reactor. In the model, fresh medium containing an identical concentration of metabolites  $S_1$  and  $S_2$  is continuously added into the reactor at a constant rate. The culture is also continuously drained at the same rate. Both types of bacteria generate the metabolite  $S_2$  from the metabolite  $S_1$ . Both metabolites diffuse in or out of bacteria through their membrane and the direction of diffusion depends on the difference in metabolic concentrations. In other words, metabolites will diffuse out of the bacteria if the metabolic concentration inside of the bacteria is larger than outside (in the reactor), while metabolites will diffuse in if the metabolic concentrations are larger outside. The mathematical representation of this model is as follows in eq. 21 and 22.

$$\frac{ds_1^{(i)}}{dt} = \frac{P}{V}(s_1^{(R)} - s_1^{(i)}) - (k_{1,2}^{(i)} + k_p)s_1^{(i)} \quad (21)$$

$$\frac{ds_2^{(i)}}{dt} = \frac{P}{V}(s_2^{(R)} - s_2^{(i)}) + k_{1,2}^{(i)}s_1^{(i)} - k_p s_2^{(i)} \quad (22)$$

From the model, it can be found that temporal dynamics of concentrations of metabolites inside bacteria are determined by (1) the differences between metabolic concentrations inside the bacteria  $s_{1,2}^{(i)}$  and those in the reactor  $s_{1,2}^{(R)}$ ; (2) the permeation coefficient of the membrane  $P$ ; (3) the metabolite conversion rate from  $S_1$  to  $S_2$  in the bacteria  $k_{1,2}^{(i)}$ ; and (4) the metabolite consumption rate  $k_p$ , which is identical for both  $S_1$  and  $S_2$ , in the bacteria.

$$\frac{ds_1^{(R)}}{dt} = D(s_1^{(0)} - s_1^{(R)}) + \sum_i X^{(i)} P(s_1^{(i)} - s_1^{(R)}) \quad (23)$$

$$\frac{ds_2^{(R)}}{dt} = D(s_2^{(0)} - s_2^{(R)}) + \sum_i X^{(i)} P(s_2^{(i)} - s_2^{(R)}) \quad (24)$$

Temporal dynamics of concentrations of metabolites outside bacteria (in the reactor) in eq. 23 and 24 are determined by (1) the differences between metabolic concentrations in the reactor  $s_{1,2}^{(R)}$  and those in added fresh medium  $s_{1,2}^{(0)}$ ; (2) the resultant dilution rate  $D$ ; (3) the differences between metabolic concentrations inside the bacteria  $s_{1,2}^{(i)}$  and those in the reactor  $s_{1,2}^{(R)}$ ; (4) the permeation coefficient of the membrane  $P$ ; and (5) the population of bacteria in the reactor  $X^{(i)}$ . When this symbiosis system is stabilized, the concentrations of metabolite  $S_2$  of the reactor could be smaller than, larger than, or in between those of both bacterial strains, depending on the settings of the above six model parameters. The authors pointed out that because the model only contains two types of metabolites, its

ability to represent competing relationships amongst complex systems is limited.

The biological symbiosis model was proven to be effective in coordinating networks with different service rates and resource consumptions coexisting by mutual interactions [5]. Morimoto et al. applied the model to the design of a portal server, which enables different P2P file-sharing networks to coexist and cooperate. In other words, this portal server provides users with transparent utilization of multiple P2P file-sharing networks by handling search requests and shared files in place of users. The system regards the portal server as the reactor, P2P file-sharing networks as bacteria (if there are  $n$  P2P overlay networks, the model will be concerned with  $n$  types of bacteria strains), requests as metabolite  $S_1$  and files as metabolite  $S_2$ . The temporal dynamics of concentrations of metabolites (requests and files) in the reactor (the portal server) and ones in the bacteria (P2P overlay networks) are calculated according to the biological symbiosis model. The portal server interacts with P2P file-sharing networks based on the value of their own concentrations of requests and files. When the portal server's concentration of  $S_2$  (files) is smaller than the one of a certain P2P overlay network, it implies the P2P file-sharing network has a sufficient number of files. As a result, the portal server stops offering files to the network to prevent excessive supply. When the portal server's concentration of  $S_1$  (requests) is smaller than the one of a certain P2P overlay network, which implies that more peers are searching or downloading files, the portal server withdraws requests from the P2P file-sharing network to reduce the load. When the service rate is too small in a certain P2P overlay network, the portal server will send requests to both P2P file sharing networks and uploads files to both as well. When the service rate is too large in a P2P overlay network, the portal server will foster effective file sharing by providing a P2P file-sharing network with the small service rate with not only files registered by portal users, but also files obtained from other P2P file-sharing networks with a higher service rate. The paper performed numerical analysis on a scenario of the portal server with two overlay networks and showed that users can easily find files in these P2P file-sharing networks, which are cooperatively utilized by the portal server in terms of the hit ratio.

### C. Summary

For BIAs inspired by heterogeneous organisms, network systems that try to maintain a dynamic balance among heterogeneous components are appropriate candidates. Both the metaphor of the predator-prey relationship, which is described by the LV competition model, and the metaphor of biological symbiosis have the common feature: They describe a balanced state in the system, where none of the component groups will vanish as time progresses. In other words, each component group is able to continuously have a stable number of individuals in the system. These two algorithms have not been widely applied to computer networked systems. As a result, there is little in the literature comparing the performance difference amongst them. Therefore, we evaluate their advantages to classic approaches and their limitations and border conditions in table III.

TABLE XII  
PERFORMANCE COMPARISON AMONGST SELF-ORGANIZED ADAPTATION BIAs

	<b>Idiotypic Network</b>	<b>Pattern Formation</b>	<b>Division of Labor</b>	<b>Corpse Clustering and Brood Sorting</b>
Component Amount	Medium	Large	Large	Large
Component Type	Heterogeneous	Homogeneous	Homogeneous	Homogeneous
Component Interaction	Global	Local	None	None
Component Observation	Global	Local	Global	Local
Communication Overhead	Medium	High	Low	Low
Scalability	Low	High	Low	High

## V. FRAMING THE APPLICATION OF BIAs

We proposed an integrated framework to understand and utilize BIAs as they apply to computer networks (see figures 1, 2, 4, and 5). BIAs are grouped by their biological sources and mapped to their corresponding applicable networking problems. From these figures, it is can be seen that BIAs inspired by different biological sources would normally have different associated application fields. However, there are several networking problems that could employ algorithms inspired by different biological sources to provide solutions for similar system goals, e.g., self-organized adaptation, routing, consensus obtainment, and component balance maintenance. In this section, we first analyze these four types of networking problems to demonstrate the scheme for selecting appropriate BIAs based on characteristics of target networked systems. In addition, related work of this survey in the biologically inspired networking field is discussed.

### A. Self-organized Adaptation

The metaphors of idiotypic network, pattern formation, division of labor, and corpse clustering and brood sorting could all be applied to the self-organized adaptation of networked systems in continuously changing environments. However, systems that have different characteristics should choose different metaphors.

The systems suitable to employ the idiotypic network metaphor would normally have different configurations or components that are not necessarily large in number, but aim for the same optimization goals. These configurations and components stimulate or suppress each other to compete for more control over the system. The metaphor is appropriate for providing a negotiating mechanism to optimize the system to adapt to changing environments.

The systems suitable to employ the pattern formation metaphor would normally have a large number of homogeneous components that only perform local observation and interaction to provide an adaptive response. Each component observes the changes in the environment in its local range and tries to spread the influence of observed local environmental changes to the entire network. This makes all components collaboratively adapt to environmental changes. Based on the observed local environmental changes and the received influence, each component makes decisions on its local adaptive response for the changing environments. The reaction-diffusion algorithm associated with this metaphor is appropriate for providing self-organized control on each component to ensure that the entire network acknowledges the environmental changes through collaborative sensing. The

algorithm is able to form an emergent adaptation pattern to changes through local response of components.

Systems suitable to employ the division of labor metaphor would normally have a large number of homogeneous components. However, instead of only being able to perform local observations, each component could independently choose to perform adaptation tasks in the entire system that fit its own capability for the system's adaptation goal. The response-threshold associated with this metaphor is appropriate for providing a self-organized control on each component to optimize the work force distribution over adaptive workloads in the system. To apply this metaphor to the design of networked system control, the task must be correctly defined based on the system's adaptation goal.

The systems suitable to employ the corpse clustering and brood sorting metaphor would normally have a number of agents that randomly travel among homogeneous components. They independently pick up or drop off data generated by these components based on the similarity between the data carried by them and the data in their surrounding environment. The BM and SACA models associated with this metaphor are appropriate for providing a self-organized control on each agent to optimize the data distribution in the system under changing environments. The behavior of components in the network is guided by the data distribution, and adapt to the changing environments as a result.

Table XII summarizes the comparison among BIAs that are applicable to the self-organized adaptation problem.

### B. Routing

The metaphors of insects foraging and information epidemics could be applied to the problem of routing in networked systems. However, systems that have different degrees of connectivity should consider different metaphors. Ant and bee foraging inspired algorithms are suitable for routing in networks where simultaneous paths among different nodes are often available. Although the topology of this type of network might change continuously due to the dynamic movement of nodes, the degree of connectivity among neighboring nodes is high enough that nodes could receive timely information about their context in the network. On the contrary, information epidemics inspired algorithms are suitable for routing in disconnected or intermittently connected networks. This type of network is so dynamic that simultaneous paths among different nodes are often unavailable. Nodes are often isolated and cannot get timely information about their context in the network. As a result, resource consumption is often the focus of the insect foraging inspired routing algorithms, while the

TABLE XIII  
PERFORMANCE COMPARISON AMONGST ROUTING BIAs

	<b>Foraging</b>	<b>Epidemics</b>
Simultaneous Path Amount	Large	Small
Isolated Node Amount	Small	Large
Communication overhead	High	Low
Latency	Low	High
Storage Requirements	Low	High

success of data delivery is often the focus of the information epidemics inspired routing algorithms.

Table XIII summarizes the comparison among BIAs that are applicable to the routing problem.

### C. Consensus Obtainment

The pulse-coupled oscillator and division of labor metaphors can both be employed to achieve consensus among components in networked systems. However, systems that have components with different characteristics should choose different metaphors. The pulse-coupled oscillator inspired consensus obtainment algorithms are suitable for networked systems whose components have a limited local observation range, but a high degree of communication with their neighboring peers. These algorithms achieve consensus in the system through the components' local synchronizing observations and adjustments based on the encoding of each component's local decision into its time phase. On the contrary, the division of labor inspired consensus obtainment algorithms are suitable for networked systems whose components have global observation range and do not necessarily have much communication with their neighboring peers. These algorithms achieve consensus in the system through the components' own global observation and task selection. There is no interaction between neighboring components. Components that select the same tasks share the consensus on their decision making.

Researchers that are interested in this field can further consult the comprehensive survey paper by Olfati-Saber et al. [177]. The authors built a theoretical framework to demonstrate the connection between different consensus applications. They illustrated the relationship between spectral and structural properties of networked systems along with the speed of consensus information diffusion. In addition, the robustness of consensus algorithms have been investigated in environments with dynamically changing topology [178] and measurement noise [179].

Table XIV summarizes the comparison among BIAs that are applicable to the consensus obtainment problem.

### D. Component Balance Maintenance

The metaphor of predator-prey relationship and biological symbiosis could both be employed to maintain a dynamic balance among heterogeneous components in networked systems, so that each component group is able to continuously have a stable number of individuals in the system. However, as shown in table XI systems that have different relationship characteristics among their components should choose different metaphors. The predator-prey relationship metaphor

TABLE XIV  
PERFORMANCE COMPARISON AMONGST CONSENSUS OBTAINMENT BIAs

	<b>PCO</b>	<b>Division of Labor</b>
Observation Range	Local	Global
Component Interaction	Local	None
Communication Overhead	High	Low
Scalability	High	Low

is suitable for networked systems having heterogeneous components that compete for limited resources. The biological symbiosis metaphor is appropriate for networked systems having heterogeneous components that rely on each other. Future networking designs that employ these types of BIAs must consider the mappings from biological metaphors to networking control schemes. Both the group type of individuals and the relationship among different groups should carefully fit the components and their relationships in networked systems. The predator-prey BIA has been applied to the congestion prevention procedure in WSN. The symbiosis BIA has been applied to the resource request-respond management procedure in P2P networks.

### E. Related Work

There are several surveys on biologically inspired algorithms [180], [181], [182], [183], [184], [185], [186]. In building on this work, this paper provides a view on future applications in the computer networking field for BIAs. First, this paper is the first survey paper that groups BIAs based on characteristics of their biological sources. These characteristic-groups not only help readers to conveniently capture the mathematical nature of BIAs, but also assist readers in judging whether a system qualifies the preconditions of the candidate algorithms. In addition, specific application and future areas of work are analyzed based on the review of existing applications. This type of analysis is usually general in other surveys. Besides, limitations and border conditions of BIAs are analyzed. Other surveys are often biased on the positive aspects of BIAs. Moreover, as discussed in the above subsections, there are similar types of problems that can be solved by different algorithms. This paper is the first survey paper that demonstrates the characteristics of target systems for related BIAs, in order to help the reader to make appropriate selections among these algorithms in different application scenarios.

BIAs have not been widely applied to industrial products in the computer networking field. As demonstrated in [96] and [145], when BIAs are applied to real devices, their performance is often not as good as shown in analytical modeling and simulations. This is because assumptions made in the theoretical evaluation might fail in complex practical applications, which includes unexpected noise, impacts from other components in the system, insufficient number of participants, etc. To overcome these limits, properties of each target networked system must be inspected in order to verify that it meets preconditions for employing BIAs. For systems that are not qualified, future work is needed to modify the assumptions and corresponding algorithms.

## VI. CONCLUSIONS

In this paper, we first show how BIAs possess characteristics that are desirable for future computer networking applications by comprehensively surveying the biological sources, mathematical models, major applications, advantages to corresponding “classic” approaches, and limitations and border conditions of these algorithms, in an organized manner.

Based on that, we provide a structured analysis of the potential directions for future application of BIAs in the computer networking field. More specifically, we describe the solution space that BIAs offer by indicating the types of problems where a given algorithm might apply. The analysis is summarized in tables I, IV, XI, and III.

Additionally, we compare performance amongst each type of BIA and summarize this in tables II, V, VIII, IX, and X. BIAs that are inspired by different biological sources but are applicable to similar networking applications are compared and summarized in tables XII, XIII, and XIV.

We also provide an integrated framework to help readers better understand and utilize the application of BIAs to network systems in figures 1, 2, 4, and 5. The research field of BIAs for computer networking is full of potential to explore, and we anticipate that this paper will help to both advance and stimulate interest in this area.

## REFERENCES

- [1] N. Mitton, A. Busson, and E. Fleury, “Self-organization in large scale ad hoc networks,” in *Mediterranean ad hoc Networking Workshop (MedHocNet’04)*, Jun 2004.
- [2] D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, “A bio-inspired algorithm for energy optimization in a self-organizing data center,” in *Proc. First international conference on Self-organizing architectures*, ser. SOAR’09. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 127 – 151.
- [3] T. Henderson, R. Venkataraman, and G. Choikim, “Reaction-diffusion patterns in smart sensor networks,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, May 2004, pp. 654 – 658.
- [4] B. Atakan and O. Akan, “BIOlogically-Inspired Spectrum Sharing in Cognitive Radio Networks,” in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Mar 2007, pp. 43 – 48.
- [5] N. Wakamiya and M. Murata, “Bio-inspired analysis of symbiotic networks,” in *Proc. 20th international teletraffic conference on Managing traffic performance in converged networks (ITC)*, 2007, pp. 204 – 213.
- [6] P. Matzinger, “Tolerance, danger, and the extended family,” *Annual Review of Immunology*, vol. 12, no. 1, pp. 991 – 1045, 1994.
- [7] ———, “The danger model in its historical context,” *Scandinavian Journal of Immunology*, vol. 54, no. 1 – 2, pp. 4 – 9, 2001.
- [8] “The Danger Project,” [www.dangertheory.com/](http://www.dangertheory.com/).
- [9] J. O. Kephart, “A Biologically Inspired Immune System for Computers,” in *Artificial Life IV: Proc. Fourth International Workshop on the Synthesis and Simulation of Living Systems*. MIT Press, 1994, pp. 130 – 139.
- [10] N. K. Jerne, “The Immune System,” *Scientific American*, vol. 229, pp. 52 – 60, 1973.
- [11] J. Suzuki and Y. Yamamoto, “Building an Artificial Immune Network for Decentralized Policy Negotiation in a Communication Endsystem: OpenWebServer/iNexus Study,” in *Proc. 4th World Conference on Systemics, Cybernetics and Informatics (SCI)*, 2000.
- [12] B. Atakan and O. Akan, “Immune System-based Energy Efficient and Reliable Communication in Wireless Sensor Networks,” in *Advances in Biologically Inspired Information Systems*, ser. Studies in Computational Intelligence, F. Dressler and I. Carreras, Eds., 2007, vol. 69, pp. 189 – 209.
- [13] B. Atakan and O. B. Akan, “Immune System Based Distributed Node and Rate Selection in Wireless Sensor Networks,” in *Proc. Bio-Inspired Models of Network, Information and Computing Systems*, Dec 2006, pp. 1 – 8.
- [14] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, “Spatio-temporal correlation: theory and applications for wireless sensor networks,” *Computer Networks Journal (Elsevier)*, vol. 45, pp. 245 – 259, 2004.
- [15] J. Timmis, M. Neal, and J. Hunt, “An artificial immune system for data analysis,” *Biosystems*, vol. 55, no. 1-3, pp. 143 – 150, 2000.
- [16] R. Langman, “The ‘complete’ idiotypic network is an absurd immune system,” *Immunology Today*, vol. 7, no. 4, pp. 100 – 101, Apr 1986.
- [17] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, “Self-Nonself Discrimination in a Computer,” in *Proc. IEEE Symposium on Security and Privacy*, ser. SP ’94. IEEE Computer Society, pp. 202 – 213.
- [18] S. A. Hofmeyr and S. Forrest, “Immunity by Design: An Artificial Immune System,” in *Proc. Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, 1999, pp. 1289 – 1296.
- [19] S. A. Hofmeyr and S. A. Forrest, “Architecture for an Artificial Immune System,” *Evol. Comput.*, vol. 8, pp. 443 – 473, Dec 2000.
- [20] J.-Y. Le Boudec, , and S. Sarafijanovic, “An Artificial Immune System Approach to Misbehavior Detection in Mobile Ad-Hoc Networks,” in *Proc. Bio-ADIT*, 2004, pp. 96 – 111.
- [21] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.
- [22] M. Drozda, S. Schauft, and H. Szczepicka, “AIS for misbehavior detection in wireless sensor networks: Performance and design principles,” in *IEEE Congress on Evolutionary Computation (CEC)*, Sep 2007, pp. 3719 – 3726.
- [23] S. Sarafijanovic and J.-Y. Le Boudec, “An Artificial Immune System for Misbehavior Detection in Mobile Ad-Hoc Networks with Virtual Thymus, Clustering, Danger Signal, and Memory Detectors,” in *Artificial Immune Systems*, ser. Lecture Notes in Computer Science, G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, Eds., 2004, vol. 3239, pp. 342 – 356.
- [24] T. Stibor, P. Mohr, J. Timmis, and C. Eckert, “Is negative selection appropriate for anomaly detection?” in *Proc. conference on Genetic and evolutionary computation (GECCO)*, 2005, pp. 321 – 328.
- [25] F. González, D. Dasgupta, and J. Gómez, “The effect of binary matching rules in negative selection,” in *Proc. international conference on Genetic and evolutionary computation: PartI*, ser. GECCO’03, pp. 195 – 206.
- [26] U. Aickelin and S. Cayzer, “The Danger Theory and Its Application to Artificial Immune Systems,” in *Proc. 1st International Conference on Artificial Immune Systems*, 2002, pp. 141–148.
- [27] J. Kim and P. J. Bentley, “An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection,” in *Proc. Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, 2001, pp. 1330 – 1337.
- [28] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. Mcleod, “Danger Theory: The Link between AIS and IDS?” in *Proc. Second International Conference on Artificial Immune Systems (ICARIS)*, 2003, pp. 147 – 155.
- [29] S. M. Garrett, “How do we evaluate artificial immune systems?” *Evol. Comput.*, vol. 13, no. 2, pp. 145 – 177, Jun 2005.
- [30] D. Dasgupta, “Artificial neural networks and artificial immune systems: similarities and differences,” in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 1, Oct 1997, pp. 873 – 878.
- [31] J. D. Farmer, N. H. Packard, and A. S. Perelson, “The immune system, adaptation, and machine learning,” *Phys. D*, vol. 2, no. 1 – 3, pp. 187 – 204, Oct 1986.
- [32] J. H. Holland, “Computation & intelligence.” Menlo Park, CA, USA: American Association for Artificial Intelligence, 1995, ch. Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems, pp. 275 – 304.
- [33] W. B. Cannon, *The wisdom of the body*. W. W. Norton and Company, Inc., 1939.
- [34] A. Somayaji, “Immunology, diversity, and homeostasis: The past and future of biologically inspired computer defenses,” *Inf. Secur. Tech. Rep.*, vol. 12, pp. 228 – 234, Sep 2007.
- [35] M. Neal and J. Timmis, “Once More Unto the Breach: Towards Artificial Homeostasis?” in *Recent Developments in Biologically Inspired Computing*, L. N. D. Castro and F. J. V. Zuben, Eds. Idea Group, Jan 2005, pp. 340 – 365.
- [36] ———, “Timidity: a useful emotional mechanism for robot control?” *Informatica*, vol. 27, no. 2, pp. 197 – 204, 2003.
- [37] N. D. Owens, J. Timmis, A. J. Greensted, and A. M. Tyrell, “On immune inspired homeostasis for electronic systems,” in *Proc. the 6th international conference on Artificial immune systems*, ser. ICARIS’07, pp. 216 – 227.

- [38] A. Somayaji and S. Forrest, "Automated response using system-call delays," in *Proc. 9th conference on USENIX Security Symposium - Volume 9*. USENIX Association, 2000, pp. 14 – 14.
- [39] J. Twycross and M. M. Williamson, "Implementing and testing a virus throttle," in *Proc. 12th conference on USENIX Security Symposium - Volume 12*. USENIX Association, 2003, pp. 20 – 20.
- [40] A. Weeks, S. Stepney, and F. Polack, "F.a.c.: Neutral emergence: a proposal," in *Symposium on Complex Systems Engineering, RAND Corporation*, 2007, pp. 11 – 12.
- [41] T. Pawson, "Protein modules and signalling networks," *Nature*, vol. 373, no. 6515, pp. 573 – 580, Feb 1995.
- [42] B. Krger and F. Dressler, "Molecular Processes as a Basis for Autonomous Networking," in *IPSI Trans. Advances Research: Issues in Computer Science and Engineering*, 2005, pp. 43 – 50.
- [43] F. Dressler, I. Dietrich, R. German, and B. Krüger, "Efficient operation in sensor and actor networks inspired by cellular signaling cascades," in *Proc. 1st international conference on Autonomic computing and communication systems*, ser. Autonomics '07. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 7:1 – 7:10.
- [44] ———, "A rule-based system for programming self-organized sensor and actor networks," *Comput. Netw.*, vol. 53, pp. 1737 – 1750, Jul 2009.
- [45] C. F. Tschudin, "Fraglets - a Metabolistic Execution Model for Communication Protocols," in *Proc. 2nd Annual Symposium on Autonomous Intelligent Networks and Systems (AINS)*, Menlo Park, 2003.
- [46] C. Tschudin and L. Yamamoto, "A Metabolic Approach to Protocol Resilience," in *Autonomic Communication*, ser. Lecture Notes in Computer Science, M. Smirnov, Ed., 2005, vol. 3457, pp. 201 – 206.
- [47] F. Dressler, "Bio-inspired Feedback Loops for Self-Organized Event Detection in SANETs," in *Proc. 3rd International Workshop on Self-Organizing Systems*, ser. IWSOS '08. Springer-Verlag, pp. 256 – 261.
- [48] ———, "Self-organized event detection in sensor networks using bio-inspired promoters and inhibitors," in *Proc. 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems*, ser. BIONETICS '08, pp. 10:1 – 10:8.
- [49] L. Yamamoto, D. Schreckling, and T. Meyer, "Self-replicating and self-modifying programs in fraglets," in *Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, Dec 2007, pp. 159 – 167.
- [50] M. Petrocchi, "Crypto-fraglets: networking, biology and security," in *Proc. 1st international conference on Bio inspired models of network, information and computing systems (BIONETICS)*, 2006.
- [51] F. Martinelli and M. Petrocchi, "Access control mechanisms for fraglets," in *Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, Dec 2007, pp. 101 – 104.
- [52] ———, "Signed and weighted trust credentials in fraglets," in *Proc. 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems*, ser. BIONETICS '08. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 7:1 – 7:4.
- [53] R. Nagpal, "A catalog of biologically-inspired primitives for engineering self-organization," in *Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering, Volume 2977 of Lecture Notes in Computer Science*. Springer, 2004, pp. 53 – 62.
- [54] J. L. Fernandez-Marquez, J. L. Arcos, G. D. M. Serugendo, M. Viroli, and S. Montagna, "Description and composition of bio-inspired design patterns: the gradient case," in *Proc. 3rd workshop on Biologically inspired algorithms for distributed systems*, ser. BADS '11. New York, NY, USA: ACM, 2011, pp. 25 – 32.
- [55] G. E. Robinson, "Modulation of alarm pheromone perception in the honey bee: evidence for division of labor based on hormonally regulated response thresholds," *J. Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 160, pp. 613 – 619, 1987.
- [56] ———, "Regulation of Division of Labor in Insect Societies," *Annual Review of Entomology*, vol. 37, no. 1, pp. 637 – 665, 1992.
- [57] C. Detrain and J. M. Pasteels, "Caste differences in behavioral thresholds as a basis for polyethism during food recruitment in the ant," *J. Insect Behavior*, vol. 4, pp. 157 – 176, 1991.
- [58] ———, "Caste Polyethism and Collective Defense in the Ant, *Pheidole pallidula*: The Outcome of Quantitative Differences in Recruitment," *Behavioral Ecology and Sociobiology*, vol. 29, no. 6, pp. 405 – 412, 1992.
- [59] T. D. Seeley, "Social foraging in honey bees: how nectar foragers assess their colony's nutritional status," *Behavioral Ecology and Sociobiology*, vol. 24, pp. 181 – 199, 1989.
- [60] ———, "The tremble dance of the honey bee: message and meanings," *Behavioral Ecology and Sociobiology*, vol. 31, pp. 375 – 383, 1992.
- [61] E. Bonabeau, G. Theraulaz, and J. L. Deneubourg, "Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insect Societies," *Proceedings: Biological Sciences*, vol. 263, no. 1376, 1996.
- [62] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg, "Response Threshold Reinforcement and Division of Labor in Insect Societies," Santa Fe Institute, Tech. Rep., Jan 1998.
- [63] S. N. Beshers and J. H. Fewell, "Models of division of labor in social insects," *Annual Review of Entomology*, vol. 46, no. 1, pp. 413 – 440, 2001.
- [64] T. Labella and F. Dressler, "A Bio-Inspired Architecture for Division of Labour in SANETs," in *Advances in Biologically Inspired Information Systems*, ser. Studies in Computational Intelligence, F. Dressler and I. Carreras, Eds. Springer Berlin / Heidelberg, 2007, vol. 69, pp. 211 – 230.
- [65] K. H. Low, W. K. Leow, and M. H. Ang, "Task allocation via self-organizing swarm coalitions in distributed mobile sensor network," in *Proc. 19th national conference on Artificial intelligence*, ser. AAAI'04. AAAI Press, pp. 28 – 33.
- [66] B. Wild and K. Ramchandran, "Detecting primary receivers for cognitive radio applications," Nov 2005, pp. 124 – 130.
- [67] A. Turing, "The chemical basis of morphogenesis," *Bulletin of Mathematical Biology*, vol. 52, pp. 153 – 197, 1990.
- [68] J. D. Murray, *Mathematical Biology II: Spatial Models and Biomedical Applications*, ser. Interdisciplinary Applied Mathematics. Springer New York, 2003, vol. 18.
- [69] K. Hyodo, N. Wakamiya, E. Nakaguchi, M. Murata, Y. Kubo, and K. Yanagihara, "Experiments and considerations on Reaction-Diffusion based Pattern Generation in a Wireless Sensor Network," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun 2007, pp. 1 – 6.
- [70] G. Neglia and G. Reina, "Evaluating activator-inhibitor mechanisms for sensors coordination," in *Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, Dec 2007, pp. 129 – 133.
- [71] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, pp. 5:1 – 5:39, Feb 2009.
- [72] K. Hyodo, N. Wakamiya, and M. Murata, "Reaction-diffusion based autonomous control of camera sensor networks," in *Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, Dec 2007, pp. 97 – 100.
- [73] A. Yoshida, K. Aoki, and S. Araki, "Cooperative Control Based on Reaction-Diffusion Equation for Surveillance System," in *Knowledge-Based Intelligent Information and Engineering Systems*, ser. Lecture Notes in Computer Science, R. Khosla, R. J. Howlett, and L. C. Jain, Eds. Springer Berlin / Heidelberg, 2005, vol. 3683, pp. 176 – 176.
- [74] M. Durvy and P. Thiran, "Reaction-diffusion based transmission patterns for ad hoc networks," in *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, Mar 2005, pp. 2195 – 2205.
- [75] Y. Chen and T. C. Henderson, "S-NETS: Smart Sensor Networks," in *Experimental Robotics VII*, ser. ISER '00. Springer-Verlag, 2001, pp. 81 – 90.
- [76] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Inc., 1999.
- [77] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The dynamics of collective sorting robot-like ants and ant-like robots," in *Proc. first international conference on simulation of adaptive behavior on From animals to animats*. MIT Press, 1990, pp. 356 – 363.
- [78] E. D. Lumer and B. Faieta, "Diversity and adaptation in populations of clustering ants," in *Proc. third international conference on Simulation of adaptive behavior : from animals to animats*. MIT Press, 1994, pp. 501 – 508.
- [79] H. Gutowitz, "Complexity-Seeking Ants," in *Proc. Third European Conference on Artificial Life*, 1994.
- [80] A. L. Vizine, R. N. D. Castro, E. R. Hruschka, and R. R. Gudwin, "Towards Improving Clustering Ants: An Adaptive Ant Clustering Algorithm," *Informatica*, pp. 143 – 154, 2005.
- [81] N. Monmarché, M. Slimane, and G. Venturini, "On Improving Clustering in Numerical Databases with Artificial Ants," in *Proc. 5th European Conference on Advances in Artificial Life*, ser. ECAL '99. Springer-Verlag, pp. 626 – 635.
- [82] N. Monmarché, "On data clustering with artificial ants," in *AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions*, A. Freitas, Ed., Jul 1999, pp. 23 – 26.

- [83] V. Ramos and J. J. Merelo, "Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning," *ArXiv Computer Science e-prints*, Dec 2004.
- [84] V. Ramos, F. Muge, and P. Pina, "Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies," *Frontiers in Artificial Intelligence and Applications Soft Computing Systems Design Management and Applications*, vol. 87, pp. 500 – 509, 2002.
- [85] J. Handl, J. Knowles, and M. Dorigo, "Ant-Based Clustering and Topographic Mapping," *Artif. Life*, vol. 12, pp. 35 – 61, Jan 2006.
- [86] O. M. Jafar and R. Sivakumar, "Ant-based Clustering Algorithms: A Brief Survey," in *International Journal of Computer Theory and Engineering*, vol. 2, no. 5, 2010, pp. 787 – 796.
- [87] J. Handl and B. Meyer, "Ant-based and swarm-based clustering," *Swarm Intelligence*, vol. 1, pp. 95 – 113, 2007.
- [88] W. bin and S. Zhongzhi, "A clustering algorithm based on swarm intelligence," in *Proc. International Conferences on Info-tech and Infonet (ICII)*, vol. 3, 2001, pp. 58 – 66 vol.3.
- [89] J. Handl, J. Knowles, and M. Dorigo, "Design and application of hybrid intelligent systems." IOS Press, 2003, ch. On the performance of ant-based clustering, pp. 204 – 213.
- [90] V. Ramos and A. Abraham, "ANTIDS: Self Organized Ant-Based Clustering Model for Intrusion Detection System," in *International Workshop on Soft Computing as Transdisciplinary Science and Technology*. Springer Verlag, 2004.
- [91] A. Forestiero, C. Mastrianni, and G. Spezzano, "Antares: an ant-inspired P2P information system for a self-structured grid," in *Proc. Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS)*, Dec 2007, pp. 151 – 158.
- [92] M. Dorigo, "Optimization, Learning and Natural Algorithms," Ph.D. dissertation, Politecnico di Milano, Italy, 1992.
- [93] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive Feedback as a Search Strategy," *Technical Report No. 91-016, Politecnico di Milano, Italy*, 1991.
- [94] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artif. Life*, vol. 5, pp. 97 – 116, Apr 1999.
- [95] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 26, no. 1, pp. 29 – 41, Feb 1996.
- [96] V. Maniezzo, L. M. Gambardella, and F. D. Luigi, "Ant colony optimization," 2004.
- [97] R. Schoonderwoerd, J. L. Bruten, O. E. Holland, and L. J. M. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adapt. Behav.*, vol. 5, no. 2, pp. 169 – 207, 1996.
- [98] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *J. Artif. Int. Res.*, vol. 9, pp. 317 – 365, December 1998.
- [99] M. Ad, H. Networks, F. Ducatelle, F. Ducatelle, G. D. Caro, G. D. Caro, L. M. Gambardella, and L. M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks," *International J. Computational Intelligence and Applications (IJCIA)*, vol. 5, pp. 169–184, 2005.
- [100] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiraman, "HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network," *Ad Hoc Netw.*, vol. 7, no. 4, pp. 690 – 705, 2009.
- [101] M. Gunes, U. Sorges, and I. Bouazzi, "ARA-the ant-colony based routing algorithm for MANETs," in *Proc. International Conference on Parallel Processing Workshops*, 2002, pp. 79 – 85.
- [102] M. Roth and S. Wicker, "TERMIT: A Swarm Intelligence Routing Algorithm for Mobile Wireless Ad-Hoc Networks," *Springer SCI Series: Swarm Intelligence and Data Mining*, 2005.
- [103] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234 – 244, Oct 1994.
- [104] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mob. Netw. Appl.*, vol. 1, no. 2, pp. 183 – 197, 1996.
- [105] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," Feb 1999, pp. 90 – 100.
- [106] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in *Proc. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*. IEEE Computer Society, 1997, p. 1405.
- [107] C.-K. Toh, "Associativity-Based Routing for Ad Hoc Mobile Networks," *Wirel. Pers. Commun.*, vol. 4, no. 2, pp. 103 – 139, 1997.
- [108] Z. Haas, "A new routing protocol for the reconfigurable wireless networks," vol. 2, Oct 1997, pp. 562 – 566.
- [109] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," in *Proc. 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, 2003, pp. 303 – 314.
- [110] E. Michlmayr, "Self-organization for search in peer-to-peer networks: the exploitation-exploration dilemma," in *Proc. 1st international conference on Bio inspired models of network, information and computing systems*, ser. BIONETICS '06.
- [111] H. F. Wedde, M. Farooq, and Y. Zhang, "BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior," in *Ant Colony Optimization and Swarm Intelligence*, ser. Lecture Notes in Computer Science, M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, Eds. Springer Berlin / Heidelberg, 2004, vol. 3172, pp. 83 – 94.
- [112] K. V. Frisch, *The dance language and orientation of bees*. Harvard University Press, 1993.
- [113] H. F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat, "BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior," in *Proc. conference on Genetic and evolutionary computation*, ser. GECCO '05, pp. 153 – 160.
- [114] H. F. Wedde, M. Farooq, C. Timm, J. Fischer, M. Kowalski, M. Langhans, N. Range, C. Schletter, R. Tarak, M. Tchatcheu, F. Volmering, S. Werner, and K. Wang, "BeeAdHoc - An Efficient, Secure, Scalable Routing Framework for Mobile AdHoc Networks," School of Computer Science, University of Dortmund, Tech. Rep., 2005.
- [115] M. Farooq, *Bee-Inspired Protocol Engineering: From Nature to Networks*. Springer Publishing Company, Incorporated, 2008.
- [116] J. Biesmeijer and T. Seeley, "The use of waggle dance information by honey bees throughout their foraging careers," *Behavioral Ecology and Sociobiology*, vol. 59, pp. 133 – 142, 2005.
- [117] C. Grüter, M. S. Balbuena, and W. M. Farina, "Informational conflicts created by the waggle dance," *Proc Biol Sci*, vol. 275, no. 1640, p. 1321, 2008.
- [118] C. Grüter and W. M. Farina, "The honeybee waggle dance: can we follow the steps?" *Trends in Ecology and Evolution*, vol. 24, no. 5, pp. 242 – 247, 2009.
- [119] P. Bailis, R. Nagpal, and J. Werfel, "Positional communication and private information in honeybee foraging models," in *Proc. 7th international conference on Swarm intelligence*, ser. ANTS'10. Springer-Verlag, pp. 263 – 274.
- [120] A. Jevtic, A. Gutierrez, D. Andina, and M. Jamshidi, "Distributed bees algorithm for task allocation in swarm of robots," *IEEE Systems J.*, vol. PP, no. 99, p. 1, 2011.
- [121] M. Farooq and G. A. Di Caro, "Routing protocols for next-generation networks inspired by collective behaviors of insect societies: An overview," in *Swarm Intelligence*, ser. Natural Computing Series, C. Blum and D. Merkle, Eds. Springer Berlin Heidelberg, 2008, pp. 101 – 160.
- [122] J. Wang, B. d'Auriol, X. Wu, Y.-K. Lee, and S. Lee, "Energy consumption of swarm intelligence inspired routing algorithms in manets," in *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*, Apr 2007, pp. 803 – 806.
- [123] J. Wang and S. Lee, "A performance comparison of swarm intelligence inspired routing algorithms for manets," in *Proc. International Conference on Computational Science and Its Applications: Part II*, ser. ICCSA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 432 – 442.
- [124] S. H. Strogatz, *Sync: How Order Emerges From Chaos In the Universe, Nature, and Daily Life*. Hyperion, Apr 2004.
- [125] J. Buck, E. Buck, J. F. Case, and F. E. Hanson, "Control of flashing in fireflies," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 144, pp. 287 – 298, 1981.
- [126] E. Izhikevich, "Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 508 – 526, May 1999.
- [127] P. C. Bressloff and P. S. Coombes, "Dynamics of Strongly Coupled Spiking Neurons," *Neural Comput.*, vol. 12, no. 1, pp. 91 – 129, 2000.
- [128] C. S. Peskin, "Mathematical Aspects of Heart Physiology," Courant Institute of Mathematical Sciences, New York University, Tech. Rep., 1975.
- [129] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645 – 1662, 1990.
- [130] D. Lucarelli and I.-J. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proc. 2nd international*

- conference on Embedded networked sensor systems (SenSys)*, 2004, pp. 62 – 68.
- [131] Y.-W. Hong and A. Scaglione, “Time synchronization and reach-back communications with pulse-coupled oscillators for UWB wireless ad hoc networks,” in *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*, Nov 2003, pp. 190 – 194.
- [132] U. Ernst, K. Pawelzik, and T. Geisel, “Synchronization Induced by Temporal Delays in Pulse-Coupled Oscillators,” *Phys. Rev. Lett.*, vol. 74, no. 9, pp. 1570 – 1573, Feb 1995.
- [133] Y.-W. Hong and A. Scaglione, “A scalable synchronization protocol for large scale sensor networks and its applications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1085 – 1099, May 2005.
- [134] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, “Firefly-inspired sensor network synchronicity with realistic radio effects,” in *Proc. 3rd international conference on Embedded networked sensor systems (SenSys)*, 2005, pp. 142 – 153.
- [135] A. Tyrrell, G. Auer, and C. Bettstetter, “Fireflies as role models for synchronization in ad hoc networks,” in *Proc. 1st international conference on Bio inspired models of network, information and computing systems (BIONETICS)*, 2006, p. 4.
- [136] A.-S. Hu and S. D. Servetto, “On the scalability of cooperative time synchronization in pulse-connected networks,” *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2725 – 2748, Jun 2006.
- [137] O. Babaoglu, T. Binci, M. Jelasity, and A. Montresor, “Firefly-inspired Heartbeat Synchronization in Overlay Networks,” in *Proc. First International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, Jul 2007, pp. 77 – 86.
- [138] N. Wakamiya, S. Kashihara, and M. Murata, “A Synchronization-based Data Gathering Scheme in Unstable Radio Environments,” in *Proc. Fourth International Conference on Networked Sensing Systems (INSS)*, Jun 2007, pp. 10 – 18.
- [139] Y.-W. Hong, L. Cheow, and A. Scaglione, “A simple method to reach detection consensus in massively distributed sensor networks,” in *Proc. International Symposium on Information Theory (ISIT)*, Jun 2004, p. 250.
- [140] F. Yu, M. Huang, and H. Tang, “Biologically inspired consensus-based spectrum sensing in mobile ad hoc networks with cognitive radios,” *IEEE Network*, vol. 24, no. 3, pp. 26 – 30, May 2010.
- [141] Z. Li, F. Yu, and M. Huang, “A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 383 – 393, Jan 2010.
- [142] J. Degesys, I. Rose, A. Patel, and R. Nagpal, “DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks,” in *Proc. 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, Apr 2007, pp. 11 – 20.
- [143] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 147 – 163, Dec 2002.
- [144] A. Mutazono, M. Sugano, and M. Murata, “Evaluation of robustness in time synchronization for sensor networks,” in *Bio-Inspired Models of Network, Information and Computing Systems (Bionetics)*, Dec 2007, pp. 89 – 92.
- [145] M. Breza and J. A. McCann, “Lessons in implementing bio-inspired algorithms on wireless sensor networks,” in *Proc. 2008 NASA/ESA Conference on Adaptive Hardware and Systems*, ser. AHS ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 271 – 276.
- [146] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks,” in *Proc. 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1*, ser. NSDI’04. Berkeley, CA, USA: USENIX Association, 2004, pp. 2 – 2.
- [147] H. W. Hethcote, “The Mathematics of Infectious Diseases,” *SIAM Rev.*, vol. 42, pp. 599 – 653, Dec 2000.
- [148] J. O. Kephart and S. R. White, “Directed-Graph Epidemiological Models of Computer Viruses,” *IEEE Symposium on Security and Privacy*, vol. 0, p. 343, 1991.
- [149] C. C. Zou, W. Gong, and D. Towsley, “Code red worm propagation modeling and analysis,” in *Proc. 9th ACM conference on Computer and communications security*, ser. CCS ’02, pp. 138 – 147.
- [150] G. Kesidis, I. Hamadeh, Y. Jin, S. Jiwarsurat, and M. Vojnović, “A model of the spread of randomly scanning Internet worms that saturate access links,” *ACM Trans. Model. Comput. Simul.*, vol. 18, pp. 6:1 – 6:14, Apr 2008.
- [151] C. C. Zou, W. Gong, D. Towsley, and L. Gao, “The monitoring and early detection of internet worms,” *IEEE/ACM Trans. Netw.*, vol. 13, pp. 961 – 974, Oct 2005.
- [152] M. Vojnović and A. J. Ganesh, “On the race of worms, alerts, and patches,” *IEEE/ACM Trans. Netw.*, vol. 16, pp. 1066 – 1079, Oct 2008.
- [153] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *IEEE Commun. Mag.*, vol. 44, no. 11, pp. 134 – 141, Nov 2006.
- [154] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Duke University, Tech. Rep., Apr 2000.
- [155] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 10, pp. 477 – 486, Aug 2002.
- [156] R. Groenevelt, P. Nain, and G. Koole, “The message delay in mobile ad hoc networks,” *Perform. Eval.*, vol. 62, pp. 210 – 228, Oct 2005.
- [157] A. Al Hanbali, P. Nain, and E. Altman, “Performance of ad hoc networks with two-hop relay routing and limited packet lifetime (extended version),” *Perform. Eval.*, vol. 65, pp. 463 – 483, Jun 2008.
- [158] A. Al Hanbali, A. A. Kherani, and P. Nain, “Simple models for the performance evaluation of a class of two-hop relay protocols,” in *Proc. the 6th international IFIP-TC6 conference on Ad Hoc and sensor networks, wireless networks, next generation internet*, ser. NETWORKING’07. Springer-Verlag, pp. 191 – 202.
- [159] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05, pp. 252 – 259.
- [160] J. W. Byers, M. Luby, and M. Mitzenmacher, “A Digital Fountain Approach to Asynchronous Reliable Multicast,” *IEEE J. Sel. Areas Commun.*, vol. 20, pp. 1528 – 1540, 2002.
- [161] Y. Wang, S. Jain, M. Martonosi, and K. Fall, “Erasure-coding based routing for opportunistic networks,” in *Proc. ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05, pp. 229 – 236.
- [162] S. Jain, M. Demmer, R. Patra, and K. Fall, “Using redundancy to cope with failures in a delay tolerant network,” in *Proc. 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM ’05, pp. 109 – 120.
- [163] R. Ahlsweide, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204 – 1216, Jul 2000.
- [164] J. Widmer and J.-Y. Le Boudec, “Network coding for efficient communication in extreme networks,” in *Proc. ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05, pp. 284 – 291.
- [165] Z. J. Haas and T. Small, “A new networking model for biological applications of ad hoc sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 27 – 40, Feb 2006.
- [166] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Comput. Netw.*, vol. 51, no. 10, pp. 2867 – 2891, Jul 2007.
- [167] A. J. Lotka, *Elements of physical biology*. Williams & Wilkins company, 1925.
- [168] V. Volterra, “Variations and Fluctuations of the Number of Individuals in Animal Species living together,” *Journal du Conseil*, vol. 3, no. 1, pp. 3 – 51, 1928.
- [169] Z. Jin, M. Zhien, and H. Maoan, “The existence of periodic solutions of the n-species Lotka-Volterra competition systems with impulsive,” *Chaos, Solitons and Fractals*, vol. 22, no. 1, pp. 181 – 188, 2004.
- [170] P. Antoniou and A. Pitsillides, “A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka-Volterra competition model,” *Comput. Commun.*, vol. 33, pp. 2039 – 2047, Nov 2010.
- [171] F. Brauer and C. Castillo-Chavez, *Mathematical Models in Population Biology and Epidemiology*. Springer, Mar 2001.
- [172] P. Antoniou and A. Pitsillides, “Toward a scalable and self-adaptable Congestion Control Approach for Autonomous Decentralized Networks,” in *Nature-Inspired Smart Information Systems Annual Symposium*, Nov 2007.
- [173] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: congestion detection and avoidance in sensor networks,” in *Proc. 1st international conference on Embedded networked sensor systems*, ser. SenSys ’03, pp. 266 – 279.
- [174] S. Rangwala, R. Gummadri, R. Govindan, and K. Psounis, “Interference-aware fair rate control in wireless sensor networks,” in *Proc. conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM ’06, pp. 63 – 74.
- [175] L. Popa, C. Raiciu, I. Stoica, and D. Rosenblum, “Reducing Congestion Effects in Wireless Networks by Multipath Routing,” in *Proc. IEEE International Conference on Network Protocols*. IEEE Computer Society, 2006, pp. 96 – 105.
- [176] T. Yomo, W.-Z. Xu, and I. Urabe, “Mathematical model allowing the coexistence of closely related competitors at the initial stage of

- evolution," *Researches on Population Ecology*, vol. 38, pp. 239 – 247, 1996.
- [177] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215 – 233, Jan 2007.
- [178] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005.
- [179] M. Huang and J. Manton, "Stochastic consensus seeking with measurement noise: Convergence and asymptotic normality," in *American Control Conference, 2008*, Jun 2008, pp. 1337 – 1342.
- [180] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Comput. Netw.*, vol. 54, no. 6, pp. 881 – 900, Apr 2010.
- [181] T. Nakano, "Biologically inspired network systems: A review and future prospects," *IEEE Trans. Syst. Man, Cybern. C Appl. Rev.*, vol. 41, no. 5, pp. 630 – 643, Sep 2011.
- [182] F. Dressler, "Efficient and Scalable Communication in Autonomous Networking using Bio-inspired Mechanisms - An Overview," in *Informatica - Special Issue on Ant Colony Multi-Agent Systems*, vol. 29, 2005, pp. 183 – 188.
- [183] M. Meisel, V. Pappas, and L. Zhang, "A taxonomy of biologically inspired research in computer networking," *Comput. Netw.*, vol. 54, no. 6, pp. 901 – 916, Apr 2010.
- [184] F. Dressler and O. Akan, "Bio-inspired networking: from theory to practice," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 176 – 183, Nov 2010.
- [185] J. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, and J. Arcos, "Description and composition of bio-inspired design patterns: a complete overview," *Natural Computing*, pp. 1 – 25, May 2012.
- [186] K. Leibnitz, N. Wakamiya, and M. Murata, "Biologically inspired networking," *Cognitive Networks: Towards Self-Aware Networks*, pp. 1 – 21, 2007.



**Chenyu Zheng** is a Ph.D. candidate in Department of Computer Science, University of Colorado at Boulder, USA. He received a B.Sc. degree in Software Engineering from Fudan University, China and a B.Sc. degree in Computer Science from University College Dublin, Ireland in a joint degree program in 2008, and a M.Sc degree in Computer Science from University of Colorado at Boulder, USA in 2011. His research interests focus on biologically inspired networking and smartphone networks.



**Douglas C. Sicker** is the Chief Technology Officer and senior advisor on spectrum at the National Telecommunications and Information Administration (NTIA). Doug is also the DBC Endowed Chair in the Department of Computer Science at the University of Colorado at Boulder with a joint appointment in the Interdisciplinary Telecommunications Program. Previously, Doug served as the Chief Technology Officer of the Federal Communications Commission (FCC). Prior to this he served as a senior advisor on the FCC National Broadband Plan. Prior to this he was Director of Global Architecture at Level 3 Communications, Inc. In the late 1990s Doug served as Chief of the Network Technology Division at the FCC. Doug has also held faculty and industry positions in the field of medical sciences. Doug is a senior member of the IEEE, as well as a member of the ACM and the Internet Society. Doug has also served on numerous advisory and program committees. His research and teaching interests include network security, wireless systems and Internet policy. Doug has a BS, MS and PhD from the University of Pittsburgh.