



A taxonomy of biologically inspired research in computer networking

Michael Meisel^{a,*}, Vasileios Pappas^b, Lixia Zhang^a

^a University of California, Los Angeles, Department of Computer Science, Los Angeles, CA 90095, USA

^b IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

ARTICLE INFO

Article history:

Available online 16 December 2009

Keywords:

Bio-inspired networking
Self-organization
Emergence
Swarm intelligence
Social insect routing
Artificial immune systems
Intrusion detection
Epidemic routing

ABSTRACT

The natural world is enormous, dynamic, incredibly diverse, and highly complex. Despite the inherent challenges of surviving in such a world, biological organisms evolve, self-organize, self-repair, navigate, and flourish. Generally, they do so with only local knowledge and without any centralized control. Our computer networks are increasingly facing similar challenges as they grow larger in size, but are yet to be able to achieve the same level of robustness and adaptability. Many research efforts have recognized these parallels, and wondered if there are some lessons to be learned from biological systems. As a result, biologically inspired research in computer networking is a quickly growing field. This article begins by exploring why biology and computer network research are such a natural match. We then present a broad overview of biologically inspired research, grouped by topic, and classified in two ways: by the biological field that inspired each topic, and by the area of networking in which the topic lies. In each case, we elucidate how biological concepts have been most successfully applied. In aggregate, we conclude that research efforts are most successful when they separate biological design from biological implementation – that is to say, when they extract the pertinent principles from the former without imposing the limitations of the latter.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

In the last 15 years, we have witnessed unprecedented growth of the Internet. The tremendous size and complexity that is associated with any large-scale, distributed system is pushing the limits of our ability to manage the network, or even to fully understand its behavior. Moreover, the Internet continues to evolve at a rapid pace in order to utilize the latest technological advances and meet new usage demands. It has been a great research challenge to find an effective means to influence its future [1], and to address a number of important issues facing the Internet today, such as overall system security, routing scalability [2], effective mobility support for large numbers of moving components, and the various demands put on the network

by the ever-increasing number of new applications and devices.

Although the Internet is perhaps the world's newest large-scale, complex system, it is certainly not the first nor the only one. Certainly, the oldest large-scale, complex systems are biological. Biological systems have been evolving over billions of years, adapting to an ever-changing environment. They share several fundamental properties with the Internet, such as the absence of centralized control, increasing complexity as the system grows in size, and the interaction of a large number of individual, self-governing components, just to name a few. Despite their disparate origins (one made by nature, the other made by man), it is easy to draw analogies between these two systems. Though drawing parallels between computer systems and biology is not a new idea [3], the unprecedented complexity and scale of modern networks demands investigation from a different angle. As many researchers have argued [4–6], there is a great opportunity to find solutions in biology that can be applied to problems in networking.

* Corresponding author. Tel.: +1 415 425 9213.

E-mail addresses: meisel@cs.ucla.edu (M. Meisel), vpappas@us.ibm.com (V. Pappas), lixia@cs.ucla.edu (L. Zhang).

1.1. Topics covered

The topics that we have chosen to cover are a broad selection of those that have proven to be fruitful and have remained active in recent years. These topics are shown in Figs. 1 and 2. In Fig. 1, each topic is categorized by the biological field or fields that inspired it. The categories are those used by the US National Research Council for categorizing research programs in the life sciences [7]. Though these categories sometimes overlap, we have tried to make the best classifications possible, preferring to use categories that emphasize systems and processes over those that emphasize specific organisms (e.g., we have classified social insect routing as ethology-inspired rather than entomology-inspired). In Fig. 2, each topic is categorized by the area or areas of computer network research to which it applies.

We have chosen to omit research areas that have only an ancestral connection to biology. These research areas involve applying an abstract, biologically inspired theory or technique, but do not generally intend to draw a direct parallel between the research topic and any biological system.

One example is genetic algorithms (GA) [8]. Genetic algorithms are a well-known optimization technique which happens to have, in turn, been inspired by biology. Genetic algorithms were invented over 35 years ago, and have been applied to a diverse range of problems in computer science, many of which have nothing in common with any biological system in and of themselves.

Another notable omission is evolutionary game theory [9]. Traditionally, game theory attempts to model strategic decisions of a number of interacting individual players, where the *game* and the *players* are concepts that can be

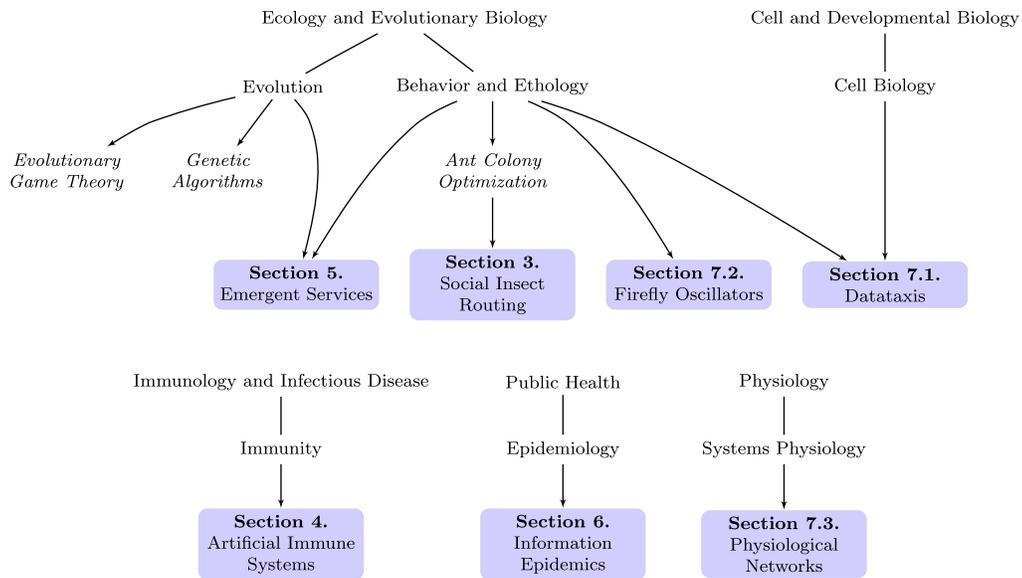


Fig. 1. A taxonomy of network research inspired by biology for the topics covered in this survey, organized by the area of inspiration. Research topics are preceded by their section number.

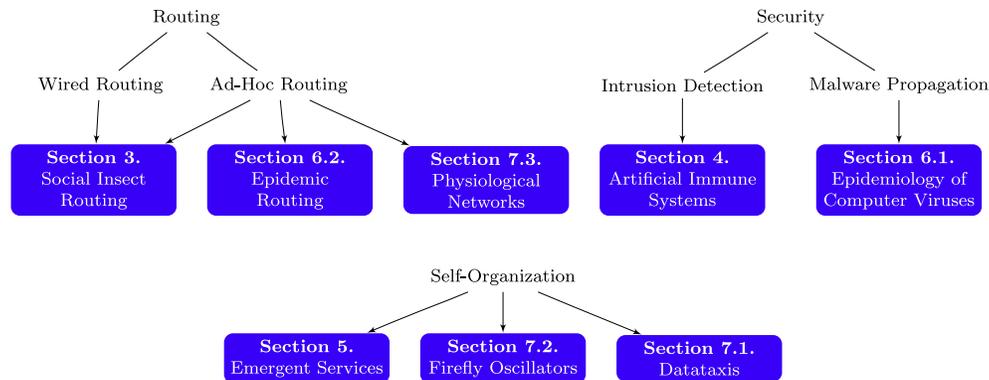


Fig. 2. An alternative taxonomy of network research inspired by biology for the topics covered in this survey, organized by the area of application. Research topics are preceded by their section number.

interpreted quite broadly. One such interpretation is to consider the game to be survival, and the players to be biological organisms. Evolutionary game theory augments traditional game theory by allowing strategies to evolve over many generations. This augmentation is crucial when modeling biological systems.

A number of research efforts have applied abstract mathematical techniques from evolutionary game theory to a wide variety of problems in networking [10–13]. However, even more so than genetic algorithms, these mathematical techniques are general enough to model a great variety of systems in a great number of disciplines, independent of any connection between the modeled system and any biological system.

1.2. Focus

In this survey, we focus on providing broad coverage of the existing literature. For each topic, we will start from a historical view, focusing on the pioneering efforts in the area. We continue with the most influential works, leading up to recent trends. Our discussion of recent trends will focus on the trendsetters, but we will also provide some examples of the most current work in the area (as of the time of writing). We frame our coverage in the context of the natural parallels between biological systems and computer networks, and provide some analysis of what makes for successful biologically inspired research. We also present some general suggestions for how to extract useful ideas and techniques from biology for use in future research.

Due to the breadth of this survey, it would be unrealistic (and probably undesirable) for our literature review to be exhaustive. Instead, wherever possible, we will include references to more exhaustive literature reviews specific to particular research topics. For readers interested in similarly broad surveys with a different focus, we recommend two excellent book chapters [14,15], both of which survey algorithms and techniques in networking that are the product of biologically inspired research.

1.3. Contents

The rest of this article is arranged as follows. Section 2 discusses in some depth why biology is an appealing and appropriate place to find inspiration for computer networking research. As shown in Figs. 1 and 2, Section 3 covers routing research inspired by the behavior of social insects, Section 4 covers intrusion and misbehavior detection research inspired by the immune system, Section 5 covers network services modeled on the interactions and evolution of populations of organisms, Section 6 covers research that applies techniques from the field of epidemiology, and Section 7 presents a sampling of newly emerging bio-inspired research topics. Section 8 mentions some other relevant research topics that we do not cover. Section 9 concludes our survey.

2. Why biology?

At first glance, it might seem a bit arbitrary to look at biology for inspiration in networking research. However,

the two fields have a much stronger connection than one might expect. The Internet – the largest, most complex, and most broadly successful computer network that exists today – has much in common with complex biological systems.

2.1. Parallel structures

In examining some of the common structures and algorithms used on the Internet today, we can find some striking similarities to biological systems. For example, anyone who has taken an introductory course in computer networks is familiar with the *hourglass model* of the Internet architecture – a diverse and rapidly changing set of applications run on top of a smaller set of transport protocols, which in turn run on a *single* Internet protocol (IP). IP runs on top of a diverse and changing set of link-layer protocols and physical mediums.

Many biological systems have a nearly identical architecture. For instance, bacteria can feed off of a variety of different nutrients. All of these nutrients contain some or all of the raw building blocks needed to power a bacterial cell. However, a bacterium must first metabolize these nutrients before they can be used, reassembling the building blocks into the multitude of complex *macromolecules* that they require for survival. Just as it would be impractical to build a different version of every Internet application for every physical-layer technology, it would be impractical for a bacterium to use a different metabolic process to convert each nutrient to each of the macromolecules it requires. Instead, much like the Internet model, the bacterial metabolism converts all nutrients to a small number of *common currencies*. These few common currencies are then used to build the large number of complex macromolecules required to power the cell. Tilting the hourglass metaphor on its side, Csete and Doyle noted that this “bow tie” structure is a nearly universal feature of complex systems [16].

Looking more deeply into the Internet protocol stack, similarities also exist at the individual protocol level. For example, at the transport layer, the Transmission Control Protocol (TCP) determines the sender’s transmission rate based on a standard congestion control algorithm. This algorithm uses feedback from the receiver, in the form of acknowledgements (ACKs), to determine when the sender’s rate should be adjusted. While the sender continues to receive the expected ACKs (positive feedback), it slowly increases its transmission speed. When the sender does *not* receive the ACKs it expects (negative feedback), it quickly reduces its transmission speed. In aggregate, this algorithm allows the sender’s rate to continuously track the optimal sending rate for the receiver, given the current state of the network. In the field of control theory, this process can be seen as a form of *integral feedback*. Long common in engineered systems, recent research has shown strong evidence that bacteria use integral feedback to govern their speed and direction of movement when tracking the concentration of certain chemicals in their environment [17].

As we discuss in Sections 3.2 and 3.3, such similarities can be found not only in the Internet protocols, but wireless protocols as well.

2.2. Complexity and robustness

Stepping back from parallel structures and algorithms, we can also find broader, systemic similarities between the Internet and biological systems. In fact, the work cited in the previous section is part of a larger attempt to develop a unified theoretical model for the structure of large-scale networks – biological, technological, and otherwise. One particular model of complex networks, termed *highly optimized tolerance* (HOT) [18], has been shown to agree with the observed structure of real-world entities – both in biological systems [19,20] and the Internet [21]. The HOT authors compare their model to other popular models of complex networks [22,23], finding that only HOT is well-aligned with the properties of real-world systems.

The theoretical claims of HOT elucidate some of the systemic similarities between the Internet and biological systems. The basic premise of HOT is that a complicated, strictly organized internal structure is necessary for any system to exhibit robust external behavior [24]. That is, there is an inherent trade-off between structural simplicity and robustness. Both the human body and the Internet have a complex, strictly organized internal structure. The human body has many different organs and physiological systems, each of which serves a specific purpose. A kidney cannot serve as a lung nor vice versa. The Internet also contains a number of specialized devices. At its core are high-speed routers, which single-mindedly forward data in a highly optimized manner. At the edges of the network are a diverse array of application-oriented devices, such as laptop computers and cellular phones. A high-speed router would be no more helpful in reading your e-mail than a kidney would be in oxygenating your blood.

Furthermore, the HOT theory claims that complex systems are only optimized to be robust against *expected* failures or perturbations. The system becomes quite fragile in regards to *unexpected* or rare failures or perturbations. For example, humans are quite robust to the sorts of changes we have evolved to tolerate – we live in climates from the Arctic to the Sahara desert, we can obtain energy from any number of different food sources, and can even survive the loss of a limb. However, a miniscule change to an important gene or exposure to trace levels of an unusual toxin can cause massive systemic failures. The Internet was optimized for robustness to physical failures of individual components, and it has proven quite successful in that regard. However, it is fragile to even small soft failures, such as an error in the design of a protocol (as occurred in the early days of the ARPAnet [25]) or a single component that breaks the rules (as with prefix hijacking [26]).

In both biological and networked systems, far simpler designs exist, but these simpler systems lack any resiliency to even the most common failures. Bacteria are composed of only a single cell, rather than a complex, structured network of cells like a human, but can only tolerate very small changes in their external environment, such as a slight change in temperature or pH. Building a network with a star topology makes many of the most difficult design challenges in the Internet, such as packet routing and addressing, trivially simple. However, a network with a star

topology is rendered completely useless by the failure of the single central node.

2.3. Biological inspirations

The evidence certainly suggests that mother nature and network engineers have not only had to solve similar problems, but have also independently converged upon strikingly similar solutions. As such, it seems entirely reasonable that new or persistent problems in computer networks could have a lot in common with problems biology has encountered and resolved long ago. As computer network researchers, it would serve us well to take a long, hard look at biological systems – we may find more answers than we expect.

3. Social insect routing

Collectively, insect societies can perform impressively complex tasks, such as nest building and food gathering. Humans tend to anthropomorphize these insects, assuming they are diligently and selflessly toiling away with the greater good of the colony in mind. The reality is far more surprising. Individual insects function much like simple computing devices – they execute simple procedures based on their input, causing them to produce some output. At any given moment, an individual insect is merely reacting to stimuli in its immediate surroundings. Large-scale, seemingly global cooperation emerges as a result of two phenomena. First, each species is genetically pre-programmed to perform an identical set of procedures given the same set of stimuli. Second, by performing these procedures, creatures implicitly modify their environment (of which they are one feature), creating new stimuli for themselves and those around them. This phenomenon of indirect communication via changes to the environment is called *stigmergy* [27,28].¹

3.1. Ant-inspired wired routing

The first and largest class of social insect-inspired routing algorithms is based on the ability of ants to converge on the shortest path from their nest to a food source. Ants accomplish this feat by laying various types of pheromone trails, or scent trails, as they travel. These pheromones serve as stimuli for other ants in their colony – ants probabilistically follow the paths with the most pheromone. When no pheromones are present, ants will follow random paths. When an ant finds a food source, it will return to the nest the same way it came. On its return journey, the ant continues to leave its pheromone trail behind, thereby increasing the amount of pheromone on the successful path. The shortest successful path will receive this extra dose of pheromone first, while the others will slowly fade away. This causes other ants to be more likely to follow the shortest path, and creates a positive feedback loop [29]. This is a classic example of stigmergy, and the basis for

¹ In the field of artificial intelligence, stigmergy is considered an example of *swarm intelligence*.

an optimization algorithm proposed by Dorigo et al. [30], which was eventually termed *ant colony optimization* (ACO).

3.1.1. ABC

The first truly stigmergy-based routing algorithm in publication was *Ant-based control* (ABC) [31], an ACO-based algorithm designed for routing in circuit-switched telephone networks. In brief, ABC makes use of active probing of the paths in the network to allow routing tables to be traffic-aware. Traffic awareness is a basic tenet of most ACO-based algorithms. It is worth exploring the ABC algorithm in some depth, as all ACO-based routing algorithms work in a similar fashion.

ABC models destinations as food sources, and routing tables are called *pheromone tables*. Each possible destination has a unique pheromone associated with it, so in an n -node network, ABC uses n different pheromones. At any node x , each entry ($d, x \rightarrow y$) in the pheromone table is the probability that an ant leaving node x will use link $x \rightarrow y$ to reach destination d . Thus, each node needs to store a pheromone level for each possible destination on each of its links, so a node with m neighbors will have a pheromone table of size $(n - 1) \times m$. This is essentially the same storage requirement as in distance vector protocols.

In order to update the pheromone tables, *ants* are sent at regular intervals from each node in the network. Ants update the pheromone table at each node they visit, but they update the entry for the source node of the ant, not the destination. To be precise, when an ant originated by node s traverses some link $p \rightarrow q$, it updates the pheromone table entry ($s, q \rightarrow p$) at node q . Thus, ABC assumes a symmetric cost on all links.

When updating pheromone tables, ABC will increase the probability of entry ($s, q \rightarrow p$) using a reinforcement learning technique. To ensure calls are set up on less congested paths, ants traveling on congested paths get delayed, and ants that have been in the network longer lay less pheromone. The amount of delay is based on the number of calls currently using the link in question. Ants select a path based on the probabilities in the pheromone table, with some random noise. Calls, however, are always set up on the links with the highest probabilities (there is no randomness involved).

One drawback specific to ABC is its winner-takes-all call setup scheme, which means that if the best route to s is congested, no call can be placed to s until the ants can change the probability distribution. ACO-based algorithms for packet-switched networks tend to exploit the multiple paths provided in the pheromone table for load balancing.

3.1.2. Packet-switched networks

Two pioneering works for ACO-based routing in packet-switched networks were developed in parallel. One was developed by Subramanian et al. [32], who adapted the ABC algorithm for use in packet-switched networks fairly faithfully. Additionally, the authors describe what they term the *uniform ant* algorithm, where ants take all paths with equal probability. This essentially amounts to a random walk, and thus has little to do with ants or stigmergy.

The other pioneering effort was by Di Caro and Dorigo, the original author of the ACO algorithm. Their algorithm is called AntNet [33]. AntNet has probably been the most influential ACO-based routing algorithm, and a number of improvements, studies, and AntNet-based algorithms have been published.² AntNet introduces the concept of *forward ants* and *backward ants*. As in ABC, forward ants stochastically follow the pheromone tables from a source to a destination. Forward ants record their path, as well as the actual time that they arrive at each node. This is how AntNet measures network congestion, since forward ants may be delayed like any other packets.

When a forward ant reaches its destination, it becomes a backward ant, and returns to the source on the reverse path of the forward ant, updating pheromone tables along the way. Pheromone levels are increased by an amount based on the time it took the forward ant to traverse each link. This forward-and-backward-ant scheme allows AntNet to support asymmetric link costs.

Unlike ABC, AntNet uses stochastic forwarding for data packets as well. This results in load balancing over multiple paths, which is one of the primary goals of AntNet. The authors' evaluation shows it to outperform a number of standard routing algorithms for constant bit rate traffic.

There are two drawbacks specific to the AntNet algorithm. It can require long delays to propagating routing information, since routing tables are only updated by backward ants. The authors do not discuss how ants acquire accurate timing information, but presumably all nodes in the network will need to have synchronized clocks for the timing information to be accurate.

In general, ACO-based routing algorithms appear advantageous for load balancing, but this feature comes at a cost. They are not guaranteed to be loop-free. The use of multiple paths concurrently may cause out-of-order packets and increased jitter in stable networks. ACO-based routing algorithms generate quite a lot of control traffic, even when the network is stable. Like ABC in circuit-switched networks, both ACO-based schemes discussed above do not deal well with link failures – before failures can be avoided, ants must first stochastically decrease the probabilities on broken links. This problem can only be avoided if nodes are able to detect link failures and explicitly notify the routing system [35].

3.2. Recent trends

Two recent trend in social insect-inspired routing are applying ACO principles to routing in mobile, ad-hoc wireless networks (MANETs) and taking inspiration from insects other than ants.

3.2.1. Ant-inspired routing for MANETs

Perhaps the most influential ant-based routing algorithm for MANETs is AntHocNet [36], a modified version of AntNet from the same authors. Since the large amount of control traffic generated by AntNet would be unacceptable in a MANET scenario, AntHocNet forgoes the use of

² See Farooq and Di Caro's 2008 survey [34] for a comprehensive review.

forward ants except for paths that are in active use. AntHocNet bears a strong resemblance to the ad-hoc on-demand distance vector (AODV) protocol [37], except with the addition of active probing of paths that have active data connections (via *proactive forward ants*), and stochastic routing of data packets over multiple paths with a distribution based on the pheromone tables. It uses hello packets to detect link failures. Unlike the original AntNet scheme, AntHocNet actually has a significant advantage in failure handling, since it can take advantage of the availability of multiple, active paths. In their evaluation, AntHocNet performs better than AODV for constant bit rate traffic, presumably due to the use of multiple paths, though variable bit rate traffic is not evaluated.

AntHocNet was not the first ant-inspired MANET routing algorithm to appear in the literature. The first appears to have been GPS/Ant-Link Algorithm (GPSAL) [38], though, along with some other early efforts [39,40], it is modeled after ants only in name – their protocols do not make use of any concepts from stigmergy.

The first MANET routing algorithm in the literature to take inspiration from ants more than just in name was Ant-Colony Based Routing Algorithm (ARA) [41]. ARA is an on-demand protocol, also resembling AODV. It waits until a node needs to send data, at which point it broadcasts AntNet-style forward ants. When a forward ant reaches the destination, the destination node prevents the forward ant from propagating further and responds with a backward ant. After this route discover phase, ARA then uses only data packets to update pheromone tables. ARA decreases all pheromone values periodically using an exponential decay factor. It also adds loop prevention and explicit failure notification. Their evaluation showed ARA's performance was slightly worse than the dynamic source routing (DSR) protocol [42], but with significantly less overhead.

3.2.2. Other insects

Increasingly, researchers have also begun exploring the behavior of other social insects as inspiration for routing algorithms. One such effort is Termite [43], a termite-inspired routing algorithm for MANETs.³

Like ants, termites also coordinate their actions via pheromones left in their environment. In fact, it was the study of termite behavior that led to the development of the original theory of stigmergy [27].

Despite the fact that the authors' inspiration came from a different insect, the Termite routing protocol [43] is quite similar to ARA [41], with two main differences. First, Termite has no loop prevention or explicit failure notification. Second, while ARA floods route requests (which ARA calls *forward ants*) and requires them to travel all the way to the destination, Termite route requests perform a random walk through the network until they discover a node with some pheromone for their destination. This triggers a route reply packet to be sent from the discovered node back to the source. This method has more in common with the

behavior of real termites than ARA's method does with real ants, but it is also less efficient. Though the Termite method could theoretically reduce control traffic volume, the authors of Termite had to include another type of control packet to spread pheromones through the network in order to compensate for its inefficiency. This particular choice may be a case where the authors used too literal a mapping from the biology to the technology.

3.2.3. Ongoing work

As of the time of writing, some researchers are still developing new and improved ant-based routing algorithms for MANETs. One example is a new algorithm by Woo et al. [46], which is based on AntHocNet. This algorithm is intended to use significantly fewer forward and backward ants, thus improving efficiency. According to their simulations, this new algorithm does, in fact, have significantly less overhead than AntHocNet while maintaining comparable performance.

Another example is HOPNET [47], which combines ant-inspired routing with the Zone Routing Protocol (ZRP) [48]. In HOPNET (as in ZRP), each node has a *zone*, which consists of all nodes within a specified number of hops from the node. Within a node's zone, HOPNET is a proactive protocol – AntNet-style forward ants are used to actively probe and maintain each path. When routing between zones, HOPNET is a reactive protocol – nodes use AntNet-style forward ants to find paths on demand. The authors compared HOPNET to AODV, ZRP, and AntHocNet in simulation with mixed results. HOPNET generally performs favorably in highly dense networks, but poorly in sparser networks.

3.3. Summary

A number of ant colony-inspired routing protocols were developed for wired networks, perhaps the most prominent of which is AntNet [33]. Though AntNet and other, similar algorithms showed promise, real implementations have not gone far. This may be due to practical considerations (some assumptions made in simulation were unrealistic), and/or simply due to the fact that the existing routing protocols are fairly mature and firmly entrenched.

Recent social insect-inspired routing research has turned to MANETs, as well as to insects other than ants. These algorithms appear quite promising, but they also tend to resemble existing MANET routing protocols. This may be due to a convergence between technology and nature. That is to say, existing MANET routing protocols, although not biologically inspired in their design, bear a strong resemblance to a deterministic version of social insect behavior. For example, AODV and DSR both send probe packets from the source to discover a roundtrip path to the destination, after which the rest of the data packets can follow that path. Compare this to ant behavior, where the first ant to make a roundtrip from nest to food source successfully marks out a path for the remaining ants.

Readers interested in other social insect-inspired routing protocols (of which there are many) should consult survey papers specific to the area [34,49,50].

³ Other notable efforts are BeeHive [44] and BeeAdHoc [45], see Section 8.

4. Artificial immune systems

One parallel between computers and biology is so well-known that it has made it into common parlance: the computer virus. The term was launched into the public vernacular in November of 1988, when Robert T. Morris, Jr. made national headlines by releasing the so-called Morris worm on the early Internet [51]. Since then, the evolution of the computer virus and other types of malicious code has coincided with the evolution of the Internet. Just as globalization has facilitated the spread of human disease, the interconnection of an increasing population of personal computers has enabled the spread of computer viruses on a global scale. This is a perfect example of a problem that was new to computer networks, but for which biology already had a solution: the immune system.

The vertebrate immune system has multiple levels of defense. The first layer of defense has a simple purpose – prevent *pathogens* (infectious agents) from entering the body in the first place. This level includes the skin, which is impenetrable to most pathogens, and bodily secretions, such as saliva, which have antibiotic properties. If a pathogen manages to breach these barriers and enter the body, it is next met by the innate immune system. The innate immune system can tell the difference between the *self* and the *foreign* – that is, cells that are part of the body and those that are not. Foreign intruders that trigger an immune response are called *antigens*. The innate immune response involves a multitude of different cellular defenders that can destroy or devour antigens. Some of these defenders, which are called *antigen presenting cells* (APCs), keep samples of the antigens they consume, and present them to the last, most advanced level of defense – the adaptive immune system.

Unlike the cells involved in the innate immune response, each cell of the adaptive immune system, which are a type of white blood cells called *lymphocytes*, will only interact with specific antigens that they are compatible with. Under the right conditions, if an APC presents one of these lymphocytes with an antigen it recognizes, the lymphocyte is sent into action. The cell rapidly clones itself, and attacks the recognized antigens directly, as well as any “self” cells infected with them. This targeted attack can be much more effective than the unspecific attacks of the innate immune system. Furthermore, when an antigen is successfully repelled, cells are programmed to remember it for an even more effective defense the next time it is encountered. Note that, like most biological systems, all of this happens with no central control. It is the aggregate behavior of many independently acting cells that is able to fight off infection.⁴

What if we were to summarize the description of the vertebrate immune system above, but replace the biological terms with networking ones? The result is as follows. First, put up firewalls and the like to prevent intrusions from getting into one’s network in the first place. Second,

in case an attacker gets in, detect and flag any suspicious activity, regardless of whether it corresponds to a known attack. Third, trigger a subsystem to examine the suspicious activity more closely, taking swift action to combat known attacks. For unknown attacks, try to figure out how to combat them effectively. When successful in combating the attack, remember the defense mechanism for next time. This sounds like a respectable high-level description for an intrusion detection system (IDS). In fact, this is the primary domain in the computer networking world where *artificial immune systems* (AIS) have been applied.

4.1. Kephart’s immune system for computers

Artificial immune systems for computers were originally conceived as a defense mechanism for individual hosts against computer viruses. In 1994, Jeffrey Kephart of IBM Research proposed one of the two earliest designs [53]. Its high-level description is similar to the immune system-based IDS description above. An analog of the innate immune response can detect an intrusion on the host via two methods: integrity checking of its programs and data, and an activity monitor that reacts to suspicious activity. However, the notion of “self” is ill-defined for a computer – users routinely modify files and install new software. Integrity checkers and activity monitors can easily be triggered by these sorts of expected behaviors. Thus, Kephart’s system does not take any defensive action at this stage. Instead, a number of so-called *decoy programs* are created. Decoy programs act a bit like APCs in the biological immune system, but instead of destroying computer viruses, they attract infection. This allows the equivalent of the adaptive response to take over, examining the virus, as well as how it infects. The goal is to find a byte sequence to serve as a signature that can be used to recognize the virus, but is unlikely to match normal programs. This is done by comparing a number of candidate signatures to a corpus of uninfected programs to determine which signature is least likely to result in false positives. This signature is then added to the database of known viruses. Known viruses can then be detected and removed using standard anti-virus techniques.

Kephart also proposed a limited extension of his system for networked machines. When a machine learned how to defend against a new virus, it would tell all of its neighbors how to do so as well. However, a neighbor would only further propagate this information if, upon learning how to detect the infection, it found that it was also infected with the same virus.

When it came to the actual implementation, Kephart took a proprietary approach. He focused on reusing code that was already in use in IBM’s AntiVirus product, as well as some code and procedures used in IBM’s virus lab. No evaluation of the system appeared in the paper. Little followup research appeared in publication, presumably because the system was destined for use in a commercial product [54]. However, much of the design and discussion in Kephart’s work foreshadowed the more detailed research in the field that would follow in the coming years.

⁴ It is worth noting that this high-level description glosses over a great amount of detail, some of which is still a matter of debate among immunologists. The interested reader can find a more complete description that is still understandable to non-biologists on Paul Bugl’s web page [52].

4.2. Negative selection

The other of the two earliest artificial immune systems was proposed by Forrest et al., published a few months prior to Kephart's work [55]. Rather than describing an entire IDS, the authors focused on integrity checking – ensuring that protected data has not been tampered with. The integrity checker is based on one specific process in the vertebrate immune system called *negative selection*.

Negative selection is the basis of much AIS research. To understand this process, we must first delve into a bit more detail about the adaptive immune system. Recall that the defender cells of the adaptive immune system are called lymphocytes. One particular type of lymphocyte, called the *killer T-cell*, is responsible for killing bodily cells that have become infected. Each T-cell has a particular set of *surface features* on its cell wall, which only allow it to bind to (and destroy) cells infected with a specific antigen. These surface features are produced pseudo-randomly when the T-cells are generated. This randomness has the advantage of allowing the system to produce T-cells that recognize antigens it has never seen before. However, it has the disadvantage that it may produce T-cells that will bind to (and destroy) healthy bodily cells. This is where negative selection comes in. Before they can enter the blood stream, T-cells are “sensitized” in the *thymus*, an organ above the heart. Negative selection is the part of this sensitizing process that destroys T-cells that would bind to healthy bodily cells.

For their integrity checking scheme, Forrest et al. created an artificial version of this process. Their system generates a fixed-size *repertoire* of random strings, deleting any strings that occur in the data that the system is told to protect. When asked to verify data, the integrity checker looks for substrings that match the strings in the repertoire. Since all “self” strings were removed from the repertoire, any match implies a verification failure.

The result is a probabilistic integrity checker with storage requirements that are independent of the size of the data set to be protected. The probability of successfully detecting an integrity failure (as well as the probabilities of false positives and false negatives) can be adjusted in two main ways: first, by adjusting the size of the repertoire, and second, by adjusting the length of substring required to constitute a match.

Forrest's Adaptive Computation Group later published the first implementation of a *network* intrusion detection system (NIDS) based on the negative selection algorithm [56] (though they were not the only researchers to propose such a system at the time [57]). This work matured into a NIDS that they called LISYS [58]. LISYS looks for intrusions by monitoring the source IP, destination IP, and port of TCP SYN packets on a LAN. The LISYS detection algorithm extends beyond negative selection into an intricate imitation of the biological immune system, a description of which is beyond the scope of this survey. Despite the increased complexity, the authors argue that each detail serves to improve the overall system defense. In simulation, LISYS appears to perform excellently, although the paper [58] does not contain any evaluation from a real-world deployment, despite the existence of an implementation.

4.3. Recent trends

Following more recent trends in networking research, AIS research has also made its way into MANETs and sensor networks in the form of misbehavior detection. In these environments, the fact that each node could have an inexperienced or nonexistent administrator means that they may require automated methods for detecting and/or avoiding malfunctioning and malicious nodes.

The other major, recent trend in AIS research is based on a relatively recent immunological theory called *the danger theory* [59]. The danger theory proposes that the immune system does not, in fact, distinguish between the self and the foreign, but rather between the safe and the dangerous. Thus, the immune system may attack self cells that appear dangerous, and may not attack foreign cells that appear safe. Of course, this difference could simply boil down to how one defines “self” and “foreign”, as Hofmeyr and Forrest suggested [58]. However, there is one functional difference between prior theory and the danger theory – the danger theory suggests the existence of a *danger signal* emitted by cells, which is a prerequisite to activation of the adaptive immune response. The theory further suggests that these signals are implicit, such as the presence of the detritus from a antigen-destroyed cell.

4.3.1. Misbehavior detection in wireless networks

Le Boudec and Sarafijanović were the first to propose an AIS for misbehavior detection in MANETs [60]. Their scheme is specifically designed to discover nodes that do not correctly implement the MANET's routing protocol; in this case, the *dynamic source routing* (DSR) protocol [42]. They used negative selection on packet traces to find nodes that performed abnormal sequences of protocol interactions. They also used an activation threshold to mitigate false positives (a technique that appeared in LISYS [58]). Their preliminary simulations showed mixed results – they found that their algorithm required a significant delay before the false positive rate fell to a reasonable level.

Drozda et al. applied a negative selection algorithm to misbehavior detection in sensor networks [61]. The authors made a number of changes to reduce the computational overhead of the algorithm. They also considered multiple levels in the protocol stack. Negative selection appeared to be somewhat less appropriate in this case, as the smaller repertoire used failed to match a number of misbehaviors.

4.3.2. Danger signals

The use of a danger signal in AIS research was popularized by Aickelin and Cayzer [62], despite a mention in Hofmeyr and Forrest's LISYS work two years earlier [58], and Burgess' notice of it two years prior still [63]. The usefulness of the danger signal concept in AIS is that suspicious activities, such as a spike in network traffic, can be used to influence the artificial immune response. It is worth noting that, without any reference to danger theory, Kephart also saw the usefulness of recognizing evidence of an attack, which he implemented in the form of activity monitors in his 1994 work [53].

In one example of an application of danger theory, Sarafijanović and Le Boudec published a followup to the work described above in which they augmented their MANET misbehavior detection scheme with a danger signal. According to their evaluation, the danger signal greatly reduced the number of false positives [64]. However, the majority of publications on the use of the danger theory in AIS have been produced by its most outspoken proponents, Aickelin et al. [62,65,66].

4.3.3. Ongoing work

A more recent paper by Sarafijanović and Le Boudec proposes an AIS for collaborative spam filtering [67]. In this scheme, each mail server runs an instance of the AIS, which generates signatures from e-mail messages using a novel technique. These signatures undergo negative selection, where signatures appearing in non-spam messages are removed. Different instances of the AIS, running on different servers, can exchange the signatures of high-volume spam messages that they have detected. The receiver treats these signatures as danger signals – it uses them to trigger more intensive/active filtering, rather than simply assuming messages with these signatures are spam. This allows an instance of the AIS to collaborate with other instances, even when they are untrusted. Their evaluation results are quite sparse, making it nearly impossible to judge the effectiveness of the system as a whole. However, the results do show that collaboration can help improve filtering accuracy.

4.4. Summary

The vertebrate immune system is so complex that even immunologists do not fully understand it. It involves multiple organs and systems, and complex interactions and co-stimulations between many different types of cells. Many of the research efforts discussed here have attempted to create artificial immune systems that closely resemble one or multiple features of the biological version, but they necessarily have an incomplete view. For example, the largest class of research efforts is based on an algorithm by Forrest et al. [55] that latched onto a particular process called *negative selection*, which plays a small part in allowing the vertebrate immune system to attack never-before-seen foreign invaders without attacking the body's own cells.

Though there is little evidence from which to judge his scheme's efficacy, Kephart's pioneering 1994 work [53] seems to touch upon most of the concepts that appear in later work. Yet, his work was the most loosely coupled to the biological implementation.

This would seem to support the idea that networking research should take advantage of its freedom from the physical constraints of biology. We may find more success in extrapolating the big ideas without constraining ourselves to following the biological implementation.

For more detail on AIS approaches to intrusion detection, the interested reader may consult Kim et al.'s excellent survey paper [68].

5. Emergent services

As the Internet has grown, so has the complexity of its application services. Today, running a popular Internet service requires the use of multiple, complex data centers scattered throughout the globe. These services, as well as their corresponding data centers, require careful and constant human design, configuration, and management. As the network continues to grow, these services will continue to increase in complexity, requiring even more human effort to keep them running.

In networks of highly mobile nodes, including embedded sensors and personal wireless devices, what application services will look like remains an open question. Mobile wireless devices in widespread use today, such as “smart” phones, rely on Internet-based application services. This service model is not appropriate for all types of wireless devices and networks, particularly when fixed infrastructure is unreliable or nonexistent. Even when fixed infrastructure is available, this service model, when combined with skyrocketing data usage among smart phone customers, has already resulted in serious scalability issues [69].

Though some existing application services for wireless networks do not rely on fixed infrastructure, they tend to be designed for a specific type of network with specific devices and configurations. In order to develop effective applications for these networks, new abstractions for the application layer are needed [70]. Furthermore, some researchers believe that the entire Internet protocol stack needs to be re-evaluated for these types of networks, resulting in new, *cross-layer* designs where the boundaries of the individual layers, including the application layer, are intentionally blurred [71].

All in all, these challenges suggest that a new paradigm for the design and implementation of application services is needed. Whether due to the increasing complexity of infrastructure-based services, or the highly dynamic, unstructured nature of infrastructure-less mobile networks, this new paradigm must allow application services to organize and configure themselves.

Complex biological systems exhibit the desirable property of self-organized emergence of beneficial, large-scale behaviors based upon the highly localized decisions of independent, autonomous components. A number of research efforts have looked to this behavior for inspiration in the design of self-organizing services.

5.1. The Bio-Networking architecture

One such research effort is the Bio-Networking Architecture project at the University of California, Irvine, headed by Tatsuya Suda. Initial work by Wang and Suda described an alternative architecture for Internet services based on the life cycle of a collection of bio-inspired *cyber-entities* [72]. Cyber-entities can reproduce, die, and migrate across the network topology, which contains multiple nodes, called *platforms*, where the cyber-entities can reside. Cyber-entities' actions consume *energy*, which they can receive by providing a service to a user, such as

servicing a web page. They expend energy when they use their platforms' resources, such as CPU power or memory, or migrate to a different platform. Wang and Suda evaluated this bio-inspired design via simulation, which showed promising but slightly mixed results. The results also seem to suggest that the system is somewhat sensitive to initial parameter values. The group later produced an implementation, which had about the same performance as the non-bio-inspired "distributed object platforms" they compared it to in their evaluation [73].

More recent work by Nakano and Suda has given cyber-entities the ability to mate, both sexually and asexually, and therefore, evolve [74]. Nakano and Suda performed a feasibility study and comparison of various initial parameter settings using a custom simulator, which showed that evolution could help mitigate the sensitivity of the system to its initial conditions. However, they did not evaluate their model's performance in comparison to other, more traditional, distributed Internet service models.

5.2. BIONETS

The BIONETS research project has similar goals and inspirations to the Bio-Networking Architecture project, but with a focus on pervasive computing rather than distributed Internet services [75,76]. The BIONETS project takes the biological organism analogy a step further by considering networks that can operate without global connectivity. The architectural details of this design were fleshed out by Carreras et al. [6] They describe a two-tiered architecture: the first tier consists of *U-nodes*, user nodes with ample storage capacity and computing power, while the second tier consists of *T-nodes*, very simple sensor nodes that can interact with *U-nodes*, but not with each other. This is a deviation from earlier models of sensor networks, where the sensor nodes communicate with each other to relay data. The *U-nodes* implement message ferrying [77] for data transit, which the authors describe as *store-carry-and-forward* – the mobility of *U-nodes* through the real world (along with their users) is the process by which data propagates through the network. No traditional routing protocol is used by either type of node, they do not even have a notion of addressing. The group has also developed a theoretical model of their system to show its feasibility for effective data transport under high-mobility conditions [78].

5.3. Recent trends

Recently, Meisel et al., have taken a somewhat different approach to self-organizing services for mobile networks [79]. In this preliminary work, the authors present a set of design guidelines derived from a careful examination of the properties exhibited by self-organizing biological systems. One unique guideline is the use of identically programmed nodes. Each node in the network runs identical algorithms that enable it to self-adapt to provide services needed by other nodes. However, this does not imply that all nodes have the same behavior at all times – rather, the nodes' program contains a set of possible *roles* for the node, only a subset of which are activated at any given time. The

program also contains the logic to adjust a node's role automatically based on the changing state of the node's environment and the overall service goals.

Furthermore, Meisel et al., take a different approach to the design process, which they refer to as a *reverse engineering* approach. They claim that other self-organizing service designs start by proposing an architecture, then observe the consequent behavior. Their reverse engineering scheme, on the other hand, involves starting with the desired results for a given service, and then trying to determine how to design the components to achieve that result.

The authors present a simulation of a design based on their guidelines. Little evaluation is provided, but the simulation shows that the design does indeed produce the desired result for the scenario considered.

5.4. Summary

Despite the various results in the area of emergent services, we believe that the research community is still in the early stages of exploring this new research direction. The efforts discussed in this section investigated the approach of building distributed systems using self-organizing entities with biologically inspired properties. Such systems are quite thought provoking, and the preliminary results show potential.

6. Information epidemics

A number of research efforts have taken notice of a parallel between the opportunistic spread of infection in human populations and the opportunistic spread of information in computer networks. These efforts are aided by the long history of mathematical epidemiology, a field which has produced a number of mathematical models for the spread of disease [80]. Two of the most common models, the *susceptible-infective-susceptible* (SIS) and *susceptible-infective-removed* (SIR) models, have been used extensively by network researchers to model the spread of information in computer networks, and are therefore worth describing in some detail.

In the SIS model, nodes are divided into two groups: the susceptible group *S* and the infective group *I*. A node from the susceptible group *S* can acquire the infection from a node in the infective group *I* with some probability. This probability can be determined in any number of ways, as we will discuss in the following section. Once a node becomes infective, it can recover from the disease, but it does not acquire any immunity, meaning it moves back to the susceptible group *S*. The recovery time is a random number that follows a certain probability distribution based on the characteristics of the disease. When a node moves from the infective group *I* to the susceptible group *S*, it can once again contract the disease from any of the remaining nodes in the infective group *I*. Thus, nodes can repeatedly move from one group to the other, and, under the right conditions, the disease may never die out.

In contrast, in the SIR model, a node cannot transition back and forth between the susceptible group *S* and the infective group *I*. Once a node contracts the disease, it

can recover from it, again after a random period of time. When a node recovers (or is removed from the population, e.g., through death), it moves to the removed group R and cannot contract the disease again. In addition, nodes in the susceptible group S cannot contract the disease from any of the nodes of the removed group R . Thus, in the SIR model, a disease will eventually die out, assuming the population of the nodes does not increase over time.

6.1. Epidemiology of computer viruses

There is one particularly obvious connection between epidemiology and computer networks: the spread of computer viruses. In 1988, just as the term “computer virus” was first entering the public vernacular, Murray spelled out this connection in detail [81]. However, with their 1991 study, Kephart and White are generally credited as the first to apply mathematical epidemiological models to the spread of computer viruses [82].

Kephart and White applied the SIS model to directed graphs where each edge can have a different probability of transmitting infection. At the time, this was quite different from the models commonly used in mathematical epidemiology. Though some models incorporated some unidirectionality between population groups or took proximity into account, the most commonly used model assumed any susceptible node had the same probability of infection. This is equivalent to a fully connected, undirected graph where all edges have the same probability of transmitting infection. In the Kephart–White directed-graph model, susceptible nodes can only become infected by an infective node if the nodes have an edge connecting them, and the directionality of the edge is from infective node to susceptible node.

Using this model, Kephart and White discovered that the topology of the directed graph (that is, the network), as well as where the infection starts on that topology, has a significant effect on how far the virus will spread. This issue has since attracted a great deal of mathematical study, which is outside the scope of this survey.⁵

Along with the rapid growth of the Internet came a similarly rapid growth in the spread of computer viruses. In 2002, Zou et al., published an in-depth analysis of the spread of one of the most famous computer viruses, the Code Red worm [86]. As a result of their investigations, the authors developed a new epidemiological model called the *two-factor worm model*. It was designed specifically to model the spread of a single worm via the Internet. This is in opposition to the Kephart–White model, which was designed to model the spread of an arbitrary number of viruses, and developed at a time when viruses primarily spread through the exchange of physical media. This meant basing the two-factor worm model on the SIR epidemiological model (instead of the SIS model) since, once a human discovers and removes the worm from a machine, they are likely to install countermeasures against a repeat infection. The “two factors” of the model refer to the two

main reasons for modifying the basic SIR model: an increase in human awareness and deployment of countermeasures over time, and a decrease in the infection rate over time due to worm-induced network congestion.

6.2. Epidemic routing

In 1987, Alan Demers et al., recognized that epidemiological models are also applicable to the spread of *desirable* information. Inspired by the SIS and SIR models, they designed *epidemic algorithms* as a method to disseminate updates throughout a distributed, replicated database [87]. Yet, it was not until 2000, once mobile, ad-hoc networks (MANETs) had begun to draw the attention of the research community, that Demers et al.’s epidemic algorithms were applied to computer network research.

There are clear topological similarities between MANETs and human networks – the mobility of nodes in a MANET is not only similar to, but often governed by, the movements of their human owners. Just as humans can only transmit infection to others within a small range of their physical location, their ad-hoc wireless devices can often only communicate with other wireless devices within a similarly small range. As a result, a MANET may never be fully connected at any particular instant in time. Yet, it is quite possible that a sender would wish to reach a destination that it does not currently have a path to. Nevertheless, routing protocols for MANETs typically only allow a sender to reach a destination if there is a path to that destination at the particular instant when the data is sent.

Vahdat and Becker developed *epidemic routing*, a routing protocol based on epidemic algorithms, as a solution to this problem [88]. As opposed to Demers et al.’s epidemic algorithms, which attempt to propagate any message to *all* nodes in the network, epidemic routing simply attempts to propagate a message to a *single* destination node. Taking advantage of node mobility, nodes opportunistically forward messages to “susceptible” nodes that happen to be in close proximity. Those nodes are then “infective,” and can carry the message towards the intended destination.

Vahdat and Becker evaluated their protocol through simulation on a set of random topologies, showing that the protocol can reach a 100% delivery rate between source and destination without unreasonable overhead. However, as discussed in Section 6.1, later results showed that the spread of an epidemic is highly dependent on the network topology. Thus, it is unclear whether these results would generalize to vastly different types of network topologies than those simulated. In fact, recent work has shown that different nodes have varying success in creating an epidemic in a mobile network [89,90].

6.3. Recent trends

With a few exceptions [89,91–93], more recent work on computer viruses and worms has focused on detection and prevention, rather than modeling their spread. This work does not focus on epidemiological models, and it is therefore outside the scope of this survey.

⁵ The interested reader may consult Section 7.2. of Newman’s comprehensive, 2003 article on complex networks [83], as well as more recent work in computer science [84,85].

Multiple variations of the basic concept of epidemic routing for MANETs have appeared in the literature in recent years [94–96]. All of them can be considered as special cases of a larger class of message delivery protocols for disconnected and intermittently connected networks, including MANETs and delay-tolerant networks (DTNs) [97]. For a detailed discussion of these message delivery protocols, we refer the interested reader to Al-Hanbali et al. [98].

6.4. Summary

The connection between epidemiology and the study of information propagation in computer networks may be the strongest of all the biological connections covered in this survey. Epidemics have long been studied through abstract, mathematical models. These models have already abstracted away the biological details, leaving computer scientists free to apply them to the study of computer networks with little modification and much success. As a result, mathematical modeling of epidemics has enabled the analytical understanding of many processes that take place in computer networks, including the spread of computer viruses and the spread of information in mobile, ad-hoc networks and delay-tolerant networks.

7. New inspirations

7.1. Datataxis

In biology, a *taxis* is the innate movement response of an organism to a stimulus. *Chemotaxis* describes the phenomenon that bacteria will innately move towards a higher concentration of certain chemicals in their environment [99]. *Datataxis* is a term coined by Lee et al. [100], to describe the movement of routed agents (vehicles, in this case) towards a higher concentration of data.

In their work, Lee et al., present datataxis as an algorithm for routing vehicular *agents* through metropolitan areas. The connection to computer networks is through vehicular ad-hoc networks (VANETs) – the agents guided by datataxis communicate wirelessly with other vehicles. The goal of the agents is to collectively harvest as much data as possible. However, this data is stored only in particular groups of moving vehicles at unknown and potentially divergent locations.

Rather than being imitative of chemotaxis alone, datataxis is inspired by multiple biological processes in combination. The agents choose a small region to harvest data from a large area based on foraging patterns of larger animals. Vehicles switch to a chemotaxis-inspired local search only when they are in an area with a data concentration above a certain threshold. Datataxis also makes use of concepts from stigmergy (see Section 3). When agents harvest data from another vehicle, they leave behind *negative pheromones* in that vehicle's data storage. If another agent attempts to harvest the same data, it will learn from the presence of the negative pheromones that this data has already been harvested, thus preventing multiple agents from uselessly harvesting the same data.

According to their simulations, the datataxis scheme proves significantly more effective than two random walk-based schemes. In fact, it is about as effective as when the agents followed a preset pattern based on an omniscient knowledge of the mobility patterns of the groups of vehicles with the highest data density.

The idea of bringing multiple biological inspirations to bear on a single research problem seems a promising one. This is one good example of how networking researchers need not apply real-world constraints when applying real-world biological concepts. Unfortunately, the authors did not evaluate the effectiveness of each of the three biologically inspired techniques in isolation, so we cannot be sure that all three are necessary in order to achieve the performance seen in their simulations. However, the authors did evaluate the effect of varying the parameters that determine when the agents switch between the animal foraging and chemotaxis modes. That evaluation showed almost no difference when the parameters were changed. The authors claim this means datataxis is robust to changes in protocol parameters. However, this could also be a sign that one of the two modes is sufficient on its own.

7.2. Firefly oscillators

Many wireless networks, particularly sensor networks, require all of their sensors to perform actions that are coordinated in time. This may be simply to synchronize duty cycles to save power, or because the sensors are measuring time-sensitive events. This is an example of a problem nature has solved – some biological systems, such as the beating of a heart and synchronized flashing of fireflies, can maintain a globally synchronous oscillation based only on local observations. In the case of fireflies, this can be over significant distances. In 1990, Mirollo and Strogatz developed a formal mathematical model of this phenomenon, which is called a *pulse-coupled oscillator* [101].

The Mirollo and Strogatz model has some limitations when applied to real-world wireless networks. In particular, it assumes no propagation delays, no lossy links, and a fully connected network. Nevertheless, it has inspired a number of synchronization schemes for wireless networks. One of the earliest of these schemes was designed for ultra-wide bandwidth wireless networks where it is feasible for each node to hear every other node [102]. This avoids one of the major limitations in applying the Mirollo and Strogatz model. The authors then modified the model to account for propagation delay and loss.

Later work by Lucarelli and Wang extended this work, showing that, in theory, the modified Mirollo and Strogatz model can be applied over multi-hop topologies [103]. Researchers at Harvard University further extended this result into an actual protocol implementation for sensor networks [104]. After some further modifications to the theoretical model, their initial results were inconclusive but promising. Upon further refinement, they were able to implement a TDMA protocol which significantly outperforms existing TDMA protocols for sensor networks, according to their evaluation [105].

7.3. Physiological networks

Physiology is the study of the internal workings of the body. At least one recent work in the area of wireless and sensor networks has developed a network structure inspired by physiological networks. Pappas et al. [106], have designed a network structure for wireless sensor networks where information flows inside the network in a way similar to the flow of blood inside the circulatory system. This constrains information to always flow in one direction on every link. No such constraint exists in most current wireless sensor network designs. Surprisingly, these circulatory system-inspired structures have a number of benefits. For example, backup links are available without the need of route recalculations, and the connectivity degree for each node is reduced, resulting in greater energy efficiency for certain types of applications.

8. Other work

As we have mentioned previously, biologically inspired network research covers a broad variety of fields and topics. Several of these topics are not covered in this survey, but still deserve mention. In this section, we provide a list of such topics, as well as some related references for the interested reader.

An emerging category of social insect-inspired routing research, which is not covered in Section 3, is bee-inspired algorithms. The two most prominent bee-inspired algorithms, both developed by Wedde et al., are BeeHive [44] and BeeAdHoc [45], the former for traditional packet-switched networks, and the latter for MANETs.

A number of research efforts have attempted to design self-organizing mobile networks based on the interaction of individual cells [107–111].

An emerging class of networks, termed *nanonetworks*, involve the communication of nanoscale devices. These networks operate at a cellular scale, and so, in fact, need to communicate in a manner that very closely resembles biological cell communication [112].

Beygelzimer et al. [113], have proposed the use of network topologies for wireless ad-hoc networks inspired by the structure of small-world social networks [114,115], which have the distinct characteristic that most of the nodes in the network are connected only with their neighboring nodes, while a small number of nodes have a few connections with very distant nodes.

9. Conclusion

A great deal of successful research in the field of computer networks has been inspired by biological systems. Yet, we believe biologically inspired networking still has much room to grow. In particular, there are great opportunities in exploring a new approach. Whether successful or not, current research tends to follow the same general philosophy:

- Observe some high-level behavior in nature which has a direct parallel to a desirable behavior for computer networks.

- Explore the basic biology of this behavior – what individual components make up the system, the processes these components perform, what mathematical models have been used to describe this behavior, and so on.
- Look for components, processes, or models that seem like they could map well to the computer networking domain.
- Turn these components, processes, or models into algorithms, new mathematical models, or software implementations. Generally attempt to stay as close as possible to the biological implementation.

This approach can and has produced intriguing and useful results, as evidenced by the research surveyed in this article. And it is understandable that this has been the dominant approach to date – it is based on the way that biologists have studied biological systems. The classical approach to biology is reductionism – study a system by breaking it down into its individual components, which are more readily amenable to rigorous scientific examination [116].

However, as others have recognized [14,117], the next generation of bio-inspired research will be most successful if it takes a more conceptual, systems-level approach. This means studying not just the behavior of individual components of the system, but their interactions, and the characteristics of the system that forms as a result. Approaches that too closely mimic the machinery of biological systems risk inheriting their quirks and constraints, imposed upon them by the randomness of evolution and the limits of the physical world. Therefore, the goal of *bio-inspired* research should be to find broader lessons and principles in the way large biological systems are built, then determine how to apply these lessons and principles to the design of networked systems. This goal requires a new high-level approach:

- Work with biologists to understand the organization and interactions of complex biological systems, from the component level all the way up to the systems level.
- Identify systems-level, organizational principles that can be applied to specific problems in the computer networking domain.
- Determine how to apply these principles to solve the problem at hand, using them to guide the development of new architectures, algorithms, and software.

Recognizing and understanding these higher-level principles requires a strong grasp of biology, as well as an awareness of current biological research. Thus, one of the major tenets of this approach is a need to work more closely with biologists. Luckily, a systems-level approach to biology, appropriately termed *systems biology*, has been gaining in popularity among biologists in recent years [116,118]. Not only can systems biologists help networking researchers to develop better biologically inspired techniques, but networking research can help them to better understand biological networks. More broadly, such collaborations can improve our understanding of the fundamental science of complex, dynamic, networked systems that underlies the two fields.

Acknowledgements

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement No. W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defence or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

- [1] C. Dovrolis, What would darwin think about clean-slate architectures?, SIGCOMM Comput Commun. Rev. 38 (1) (2008) 29–34, doi:10.1145/1341431.1341436.
- [2] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, L. Zhang, Towards a new internet routing architecture: arguments for separating edges from transit core, in: Proceedings of HotNets-VII, 2008.
- [3] J. von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, Automata Studies 34 (1956) 43–99.
- [4] J.O. Kephart, D.M. Chess, The vision of autonomic computing, Computer 36 (1) (2003) 41–50, doi:10.1109/MC.2003.1160055.
- [5] O. Babaoğlu, G. Canright, A. Deutsch, G.A. Di Caro, F. Ducatelle, L.M. Gambardella, N. Ganguly, M. Jelasity, R. Montemanni, A. Montresor, T. Urnes, Design patterns from biology for distributed computing, ACM Trans. Auton. Adapt. Syst. 1 (1) (2006) 26–66, doi:10.1145/1152934.1152937.
- [6] I. Carreras, D. Miorandi, I. Chlamtac, From biology to evolve-able pervasive ICT systems, IEEE Int. Conf. Syst. Man Cybernet. (2007) 4075–4080, doi:10.1109/ICSMC.2007.4414263.
- [7] US National Research Council, Taxonomy of fields and their subfields, July 2006 [cited June 2009]. <http://sites.nationalacademies.org/pga/Resdoc/PGA_044522>.
- [8] J.H. Holland, Genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (2) (1973) 88–105, doi:10.1137/0202009.
- [9] J. Maynard Smith, Evolution and the Theory of Games, Cambridge University Press, 1982.
- [10] Y. Zheng, Z. Feng, Evolutionary game and resources competition in the internet, in: SIBCOM-2001: The IEEE-Siberian Workshop of Students and Young Researchers, Modern Communication Technologies, 2001, pp. 51–54, doi:10.1109/SIBCOM.2001.977512.
- [11] S. Shakkottai, E. Altman, A. Kumar, Multihoming of users to access points in WLANs: a population game perspective, IEEE J. Selected Areas Commun. 25 (6) (2007) 1207–1215, doi:10.1109/JSAC.2007.070814.
- [12] E. Altman, Y. Hayel, A stochastic evolutionary game of energy management in a distributed aloha network, in: Proceedings of IEEE INFOCOM, 2008, pp. 1759–1767, doi:10.1109/INFOCOM.2008.238.
- [13] E. Altman, R. El-Azouzi, Y. Hayel, H. Tembine, The evolution of transport protocols: an evolutionary game perspective, Computer Networks 53 (10) (2009) 1751–1759, doi:10.1016/j.comnet.2008.12.023.
- [14] K. Leibnitz, N. Wakamiya, M. Murata, Biologically inspired networking, in: Cognitive Networks: Towards Self-Aware Networks, Wiley-Interscience, 2007, pp. 1–21 (Chapter 1).
- [15] F. Dressler, Bio-inspired networking – self-organizing networked embedded systems, in: Organic Computing, Understanding Complex Systems, Springer, 2008, pp. 285–302, doi:10.1007/978-3-540-77657-4.
- [16] M. Csete, J. Doyle, Bow ties, metabolism and disease, Trend Biotechnol. 22 (9) (2004) 446–450.
- [17] T.M. Yi, Y. Huang, M.I. Simon, J. Doyle, Robust perfect adaptation in bacterial chemotaxis through integral feedback control, Proc. Natl. Acad. Sci. USA 97 (9) (2000) 4649–4653.
- [18] J.M. Carlson, J. Doyle, Highly optimized tolerance: a mechanism for power laws in designed systems, Phys. Rev. E 60 (2) (1999) 1412–1427, doi:10.1103/PhysRevE.60.1412.
- [19] T. Zhou, J.M. Carlson, J. Doyle, Mutation, specialization, and hypersensitivity in highly optimized tolerance, Proc. Natl. Acad. Sci. USA 99 (4) (2002) 2049–2054, doi:10.1073/pnas.261714399.
- [20] R. Tanaka, Scale-rich metabolic networks, Phys. Rev. Lett. 94 (16) (2005) 168101, doi:10.1103/PhysRevLett.94.168101.
- [21] J.C. Doyle, D.L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, W. Willinger, The ‘robust yet fragile’ nature of the Internet, Proc. Natl. Acad. Sci. USA 102 (41) (2005) 14497–14502, doi:10.1073/pnas.0501426102.
- [22] P. Bak, C. Tang, K. Wiesenfeld, Self-organized criticality, Phys. Rev. A 38 (1) (1988) 364–374, doi:10.1103/PhysRevA.38.364.
- [23] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512, doi:10.1126/science.286.5439.509.
- [24] J.M. Carlson, J. Doyle, Complexity and robustness, Proc. Natl. Acad. Sci. USA 99 (Suppl. 1) (2002) 2538–2545, doi:10.1073/pnas.012582499.
- [25] R. Perlman, Myths, missteps, and folklore in protocol design, in: Proceedings of USENIX, 2001.
- [26] M. Lad, R. Oliveira, B. Zhang, L. Zhang, Understanding resiliency of internet topology against prefix hijack attacks, in: Proceedings of IEEE International Conference on Dependable Systems and Networks, 2007, pp. 368–377, doi:10.1109/DSN.2007.95.
- [27] P.-P. Grassé, La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: essai d’interprétation du comportement des termites constructeurs, Insectes Sociaux 6 (1) (1959) 41–80, doi:10.1007/BF02223791.
- [28] G. Theraulaz, E. Bonbeau, A brief history of stigmergy, Artificial Life 5 (2) (1999) 97–116, doi:10.1162/106454699568760.
- [29] S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels, Self-organized shortcuts in the argentine ant, Naturwissenschaften 76 (12) (1989) 579–581.
- [30] M. Dorigo, V. Maniezzo, A. Coloni, Positive feedback as a search strategy, Technical report, Politecnico di Milano, Italy, 1991.
- [31] R. Schoonderwoerd, J.L. Bruten, O.E. Holland, L.J.M. Rothkrantz, Ant-based load balancing in telecommunications networks, Adapt. Behav. 5 (2) (1996) 169–207, doi:10.1177/105971239700500203.
- [32] D. Subramanian, P. Druschel, J. Chen, Ants and reinforcement learning: a case study in routing in dynamic networks, in: Proceedings of IJCAI, Morgan Kaufmann, 1997, pp. 832–838.
- [33] G.A. Di Caro, M. Dorigo, AntNet: distributed stigmergetic control for communications networks, J. Artif. Intell. Res. 9 (1998) 317–365.
- [34] M. Farooq, G.A. Di Caro, Routing protocols for next-generation networks inspired by collective behaviors of insect societies: An overview, in: Swarm Intelligence, Natural Computing, Springer, 2008, pp. 101–160.
- [35] B. Baran, R. Sosa, A new approach for AntNet routing, in: Proceedings of IEEE ICCCN, 2000, pp. 303–308, doi:10.1109/ICCCN.2000.885506.
- [36] G. Di Caro F. Ducatelle L.M. Gambardella, AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks, in: PPSN VIII, 2004, pp. 461–470.
- [37] C. Perkins, E. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of IEEE WMCSA, 1999, pp. 90–100, doi:10.1109/MCSA.1999.749281.
- [38] D. Câmara, A. Loureiro, A GPS/ant-like routing algorithm for ad hoc networks, in: Proceedings of IEEE WCNC, vol. 3, 2000, pp. 1232–1236, doi:10.1109/WCNC.2000.904807.
- [39] H. Matsuo, K. Mori, Accelerated ants routing in dynamic networks, in: ACIS-SNPD, 2001, pp. 333–339.
- [40] S. Marwaha, C.K. Tham, D. Srinivasan, Mobile agents based routing protocol for mobile ad hoc networks, in: Proceedings of IEEE GLOBECOM, 2002, pp. 163–167.
- [41] M. Güneş, U. Sorges, I. Bouazizi, ARA – the ant-colony based routing algorithm for MANETs, in: Proceedings of IEEE ICPPW, IEEE Computer Society, Los Alamitos, CA, USA, 2002, pp. 79–85, doi:10.1109/ICPPW.2002.1039715.
- [42] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, Mobile Comput. (1996) 153–181, doi:10.1007/978-0-585-29603-6.
- [43] M. Roth, S. Wicker, Termite: emergent ad-hoc networking, in: Proceedings of 2nd Mediterranean Workshop on Ad-Hoc Networks, 2003.
- [44] H.F. Wedde, M. Farooq, Y. Zhang, Beehive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior, in: Ant Colony, Optimization, and Swarm Intelligence, LNCS, vol. 3172, Springer, 2004, pp. 83–94.

- [45] H.F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, R. Jeruschkat, BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior, in: Proceedings of GECCO, ACM, 2005, pp. 153–160. doi:10.1145/1068009.1068034.
- [46] M. Woo, N.H. Dung, W.J. Roh, An efficient ant-based routing algorithm for manets, in: Proceedings of ICACT 2008: 10th International Conference on Advanced Communication Technology, vol. 2, 2008, pp. 933–937. doi:10.1109/ICACT.2008.4493920.
- [47] J. Wang, E. Osagie, P. Thulasiraman, R.K. Thulasiram, HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network, Ad Hoc Networks 7 (4) (2009) 690–705, doi:10.1016/j.adhoc.2008.06.001.
- [48] Z.J. Haas, A new routing protocol for the reconfigurable wireless networks, in: IEEE 6th International Conference on Universal Personal Communications Record, vol. 2, 1997, pp. 562–566. doi:10.1109/ICUPC.1997.627227.
- [49] K.M. Sim, W.H. Sun, Ant colony optimization for routing and load-balancing: survey and new directions, IEEE Trans. Syst. Man Cybernet. Part A 33 (5) (2003) 560–572, doi:10.1109/TSMCA.2003.817391.
- [50] H.F. Wedde, M. Farooq, A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks, J. Syst. Architect. 52 (8) (2006) 461–484, doi:10.1016/j.sysarc.2006.02.005.
- [51] M. Wines, A Youth's Passion for Computers, Gone Sour, The New York Times, November 11, 1988.
- [52] P. Bugl, Immune system, March 2001 [cited Nov 2008]. <<http://uhaweb.hartford.edu/bugl/immune.htm>>.
- [53] J.O. Kephart, A biologically inspired immune system for computers, in: Proceedings of Artificial Life IV, MIT Press, 1994, pp. 130–139.
- [54] S. White, M. Swimmer, E. Pring, W. Arnold, D. Chess, J. Morar, Anatomy of a commercial-grade immune system, in: IBM Research White Paper. <<http://www.research.ibm.com/antivirus/SciPapers/White/Anatomy/anatomy.html>>.
- [55] S. Forrest, A. Perelson, L. Allen, R. Cherukuri, Self-nonsel discrimination in a computer, in: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, 1994, pp. 202–212. doi:10.1109/RISP.1994.296580.
- [56] S. Hofmeyr, S. Forrest, Immunity by design: an artificial immune system, in: Proceedings of Genetic and Evolutionary Computation Conference (GECCO), vol. 2, 1999, pp. 1289–1296.
- [57] J. Kim, P. Bentley, The human immune system and network intrusion detection, in: 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT), 1999.
- [58] S.A. Hofmeyr, S. Forrest, Architecture for an artificial immune system, Evolution. Comput. 8 (4) (2000) 443–473, doi:10.1162/106365600568257.
- [59] P. Matzinger, Tolerance, danger, and the extended family, Ann. Rev. Immunol. 12 (1) (1994) 991–1045, doi:10.1146/annurev.iy.12.040194.005015.
- [60] J.-Y. Le Boudec, S. Sarafijanović, An artificial immune system approach to misbehavior detection in mobile ad hoc networks, in: Proceedings of Bio-ADIT, 2004, pp. 396–411.
- [61] M. Drozda, S. Schaut, H. Szczerbicka, Ais for misbehavior detection in wireless sensor networks: performance and design principles, in: Proceedings of IEEE Congress on Evolutionary Computation, 2007, pp. 3719–3726. doi:10.1109/CEC.2007.4424955.
- [62] U. Aickelin, S. Cayzer, The danger theory and its application to artificial immune systems, in: Proceedings of 1st International Conference on Artificial Immune Systems (ICARIS 2002), 2002, pp. 141–148.
- [63] M. Burgess, Computer immunology, in: Proceedings of LISA-XII, 1998.
- [64] S. Sarafijanović, 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, Artif. Immune Syst. (2004) 342–356.
- [65] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, J. McLeod, Danger theory: the link between AIS and IDS?, Artif Immune Syst. (2003) 147–155.
- [66] Danger theory project [online].
- [67] S. Sarafijanović, J.-Y. Le Boudec, Artificial immune system for collaborative spam filtering, in: Proceedings of NISCO 2007, 2008, pp. 39–51. doi:10.1007/978-3-540-78987-1.
- [68] J. Kim, P. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, J. Twycross, Immune system approaches to intrusion detection – a review, Natural Comput. 6 (4) (2007) 413–466, doi:10.1007/s11047-006-9026-4.
- [69] M. Peers, Demands on network are an iPhone hang-up, 2009. <<http://online.wsj.com/article/SB124200303430005275.html>>.
- [70] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (4) (2007) 921–960, doi:10.1016/j.comnet.2006.10.002.
- [71] V. Srivastava, M. Motani, Cross-layer design: a survey and the road ahead, IEEE Commun. Mag. 43 (12) (2005) 112–119, doi:10.1109/MCOM.2005.1561928.
- [72] M. Wang, T. Suda, The bio-networking architecture: a biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications, In: IEEE/IPSJ International Symposium on Applications and the Internet, 2001, p. 43, doi:10.1109/SAINT.2001.905167.
- [73] J. Suzuki, T. Suda, Design and implementation of a scalable infrastructure for autonomous adaptive agents, in: Proceedings of 15th IASTED International Conference on Parallel and Distributed Computing and Systems, 2003.
- [74] T. Nakano, T. Suda, Self-organizing network services with evolutionary adaptation, IEEE Trans. Neural Networks 16 (5) (2005) 1269–1278, doi:10.1109/TNN.2005.853421.
- [75] I. Chlamtac, I. Carreras, H. Woesner, From internets to bionets: biological kinetic service oriented networks, Adv. Pervasive Comput. Networking (2005) 75–95, doi:10.1007/0-387-23466-7.
- [76] I. Carreras, I. Chlamtac, H. Woesner, C. Kiraly, BIONETS: BIO-inspired Next generation networkS, Autonomic Commun. (2005) 245–252.
- [77] M. Mukarram Bin Tariq, M. Ammar, E. Zegura, Message ferry route design for sparse ad hoc networks with mobile nodes, in: Proceedings of 7th ACM International Symposium on Mobile Ad hoc Networking and Computing, ACM, New York, NY, USA, 2006, pp. 37–48. doi:10.1145/1132905.1132910.
- [78] I. Carreras, I. Chlamtac, F. De Pellegrini, D. Miorandi, Bionets: bio-inspired networking for pervasive communication environments, IEEE Trans. Vehicular Technol. 56 (1) (2007) 218–229, doi:10.1109/TVT.2006.883762.
- [79] M. Meisel, V. Pappas, P. Zerfos, L. Zhang, Emergent mobile services, Technical Report 090015, UCLA Computer Science Department, June 2009.
- [80] H.W. Hethcote, The Mathematics of Infectious Diseases, vol. 42, 2000, pp. 599–653.
- [81] W.H. Murray, The application of epidemiology to computer viruses, Computers Security 7(2) (1988) 139–145. doi:10.1016/0167-404(88)90327-6.
- [82] J. Kephart, S. White, Directed-graph epidemiological models of computer viruses, in: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, 1991, pp. 343–359. doi:10.1109/RISP.1991.130801.
- [83] M.E.J. Newman, The structure and function of complex networks, SIAM Review 45.
- [84] A. Ganesh, L. Massoulié, D. Towsley, The effect of network topology on the spread of epidemics, in: Proceedings of IEEE INFOCOM, vol. 2, 2005, pp. 1455–1466.
- [85] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks, ACM Trans. Inf. Syst. Secur. 10 (4) (2008) 1–26, doi:10.1145/1284680.1284681.
- [86] C.C. Zou, W. Gong, D. Towsley, Code red worm propagation modeling and analysis, in: CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security, 2002, pp. 138–147. doi:10.1145/586110.586130.
- [87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: PODC '87: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, 1987, pp. 1–12. doi:10.1145/41840.41841.
- [88] A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, Technical Report CS-2000-06, Duke University, 2000. <<http://www.cs.duke.edu/vahdat/ps/epidemic.pdf>>.
- [89] J.W. Mickens, B.D. Noble, Modeling epidemic spreading in mobile environments, in: Proceedings of 4th ACM Workshop on Wireless Security, 2005, pp. 77–86. doi:10.1145/1080793.1080806.
- [90] I. Carreras, D. Miorandi, G.S. Canright, K. Engo-Monsen, Understanding the spread of epidemics in highly partitioned mobile networks, in: BIONETICS '06: Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems, 2006. doi:10.1145/1315843.1315846.
- [91] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms, in: Proceedings of IEEE INFOCOM, vol. 3, 2003, pp. 1890–1900.

- [92] Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos, Epidemic spreading in real networks: an eigenvalue viewpoint, 2003, pp. 25–34.
- [93] G. Kesidis, I. Hamadeh, Y. Jin, S. Jivasurat, M. Vojnović, A model of the spread of randomly scanning internet worms that saturate access links, *ACM Trans. Model. Comput. Simul.* 18 (2) (2008) 1–14, doi:10.1145/1346325.1346327.
- [94] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: *MobiHoc '04: Proceedings of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing*, ACM, New York, NY, USA, 2004, pp. 187–198. doi:10.1145/989459.989483.
- [95] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: *WDTN '05: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, ACM, New York, NY, USA, 2005, pp. 252–259. doi:10.1145/1080139.1080143.
- [96] R. Groenevelt, P. Nain, G. Koole, Message delay in manet, in: *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ACM, New York, NY, USA, 2005, pp. 412–413. doi:10.1145/1064212.1064280.
- [97] K. Fall, A delay-tolerant network architecture for challenged internets, in: *Proceedings of ACM SIGCOMM*, ACM, 2003, pp. 27–34. doi:10.1145/863955.863960.
- [98] A. Al-Hanbali, M. Ibrahim, V. Simon, E. Varga, I. Carreras, A survey of message delivery protocols in MANETs, in: *Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer and Communication Systems*, 2008.
- [99] J. Adler, W.-W. Tso, “Decision”-making in bacteria: chemotactic response of *escherichia coli* to conflicting stimuli, *Science* 184 (4143) (1974) 1292–1294, doi:10.1126/science.184.4143.1292.
- [100] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Lió, K.-W. Lee, Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment, *Ad Hoc Networks* 7 (4) (2009) 725–741, doi:10.1016/j.adhoc.2008.03.009.
- [101] R. Mirollo, S. Strogatz, Synchronization of pulse-coupled biological oscillators, *SIAM J. Appl. Math.* 50 (6) (1990) 1645–1662.
- [102] Y.-W. Hong, A. Scaglione, Time synchronization and reach-back communications with pulse-coupled oscillators for uwb wireless ad hoc networks, in: *IEEE Conference on Ultra Wideband Systems and Technologies*, 2003, pp. 190–194. doi:10.1109/UWBST.2003.1267830.
- [103] D. Lucarelli, I.-J. Wang, Decentralized synchronization protocols with nearest neighbor communication, in: *Proceedings of 2nd International Conference on Embedded Networked Sensor Systems*, ACM, 2004, pp. 62–68. doi:10.1145/1031495.1031503.
- [104] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, R. Nagpal, Firefly-inspired sensor network synchronicity with realistic radio effects, in: *Proceedings of SenSys*, 2005, pp. 142–153.
- [105] J. Degeys, I. Rose, A. Patel, R. Nagpal, DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks, in: *6th International Symposium on Information Processing in Sensor Networks*, 2007, pp. 11–20. doi:10.1109/IPSNS.2007.4379660.
- [106] V. Pappas, D. Verma, B.-J. Ko, A. Swami, A circulatory system approach for wireless sensor networks, *Ad Hoc Networks*. doi:10.1016/j.adhoc.2008.04.009.
- [107] S. George, D. Evans, S. Marchette, A biological programming model for self-healing, in: *Proceedings of ACM Workshop on Survivable and Self-regenerative Systems*, 2003, pp. 72–81. doi:10.1145/1036921.1036929.
- [108] B. Krüger, F. Dressler, Molecular processes as a basis for autonomous networking, in: *Symposium on Challenges in the Internet and Interdisciplinary Research (IPSI)*, 2004.
- [109] I. Wokoma, L. Shum, L. Sacks, I. Marshall, A biologically-inspired clustering algorithm dependent on spatial data in sensor networks, in: *Proceedings of the Second European Workshop on Wireless Sensor Networks*, 2005, pp. 386–390. doi:10.1109/EWSN.2005.1462030.
- [110] F. Dressler, B. Krüger, G. Fuchs, R. German, Self-organization in sensor networks using bio-inspired mechanisms, in: *Proceedings of ACM/GI/ITG ARCS Workshop on Self-Organization and Emergence*, 2005, pp. 139–144.
- [111] F. Dressler, I. Dietrich, R. German, B. Krüger, Efficient operation in sensor and actor networks inspired by cellular signaling cascades, in: *Proceedings of Autonomics*, 2007, pp. 1–10.
- [112] I. Akyildiz, F. Brunetti, C. Blázquez, Nanonetworks: a new communication paradigm, *Computer Networks* 52 (12) (2008) 2260–2279, doi:10.1016/j.comnet.2008.04.001.
- [113] A. Beygelzimer, A. Kershenbaum, K.-W. Lee, V. Pappas, The benefits of directional antennas in heterogeneous wireless ad-hoc networks, in: *Fifth IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2008, pp. 442–449.
- [114] J. Travers, S. Milgram, An experimental study of the small world problem, *Sociometry* 32 (4) (1969) 425–443.
- [115] D.J. Watts, S.H. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature* 393 (6684) (1998) 440–442, doi:10.1038/30918.
- [116] D. Noble, *The Music of Life: Biology Beyond Genes*, Oxford University Press, 2008.
- [117] J. Timmis, M. Amos, W. Banzhaf, A.M. Tyrrell, “Going back to our roots”: second generation bio-computing, *Int. J. Unconventional Comput.* 2 (4) (2006) 349–378.
- [118] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, Mathematical and Computational Biology, Chapman & Hall/CRC, 2006.



Michael Meisel is a Ph.D. student in Computer Science at the University of California, Los Angeles. He is a member of the Internet Research Lab, studying under Professor Lixia Zhang. He also holds the position of Senior Software Engineer at Large for Inveneo, a non-profit that develops and deploys information and communications technology for rural and remote areas in the developing world, where he worked full-time from 2005 to 2006. He received his B.A. in Computer Science from the University of California, Berkeley in 2003.



Vasileios Pappas is a research staff member of IBM's T J Watson Research Lab at Hawthorne NY. Before joining IBM Research he was a graduate student at the Computer Science department of University of California, Los Angeles, where he received his Ph.D. and M.Sc. degrees in 2006 and 2003, respectively. He did his undergrad at the National Technical University of Athens where he received his B.Eng. degree in 2001 from the Department of Electrical and Computer Engineering.



Lixia Zhang received her Ph.D in Computer Science from the Massachusetts Institute of Technology. She was a member of the research staff at the Xerox Palo Alto Research Center before joining the faculty of UCLA's Computer Science Department in 1995. In the past she has served as the vice chair of ACM SIGCOMM, editor for the IEEE/ACM Transactions on Networking, and member of the Internet Architecture Board. She is currently co-chair of the Routing Research Group under IRTF. She is a fellow of ACM and a fellow of IEEE.