# 1

# Introduction

During recent years, telecommunication networks have become a special focus of research, both in academia and industry [91, 93, 205]. This is certainly due to the unprecedented growth of the Internet during the last decade of the previous century as it developed into a nerve center of the communication infrastructure [168]. One important reason for the success of the Internet is its connectionless packet-switching technology (no connection is established between a sender and a receiver). Such a paradigm results in a simple, flexible, scalable and robust network layer architecture [135, 19, 189]. This is in contrast to traditional connection-oriented telecommunication networks in which a circuit is reserved for a connection between a sender and a receiver [91, 93, 205].

The Internet's success motivated researchers to realize the dream of *Ubiquitous Computing*, including the concept of "one person–many computers" [276, 278, 277, 279]. Research and development in *Ubiquitous Computing* resulted in an exponential growth of smart handheld computing devices, which have to be interconnected and connected to the Internet to satisfy highly demanding users. In turn, these requirements resulted in a phenomenal growth in wireless telecommunication networks and their supporting Internet Protocol (IP) (the standard protocol for the network layer of the Internet) on wireless networks. However, these wireless networks require an infrastructure (base station) for providing connectivity to mobile terminals. As a result, work on Mobile Ad Hoc Networks (MANETs) has become a vigorous effort. Here mobile terminals communicate with one another without the need for a communication infrastructure. These networks have turned useful or even indispensable in search and rescue operations, disaster relief management, and military command and control.

*Ubiquitous Computing* has created a demanding community of users, who are utilizing its potential in novel applications like the World Wide Web (WWW), Computer Supported Collaborative Work (CSCW), e-commerce, tele-medicine, and e-learning. An essential feature of most of these applications is the ability to transmit audio and video streams to the participants under some Quality of Service (QoS) constraints. The users want all of these services on their desktops as well as on their mobile terminals. Such challenging requirements can only be met if a network's resources are utilized in an efficient manner.

The efficient utilization of limited network resources and infrastructures by enhancing/optimizing the performance of operational IP networks is defined as *Traffic Engineering* [15, 167]. Its goals are accomplished by devising efficient and reliable routing strategies. The important features and characterizations of such routing protocols are: *loadbalancing, constraint-based routing, multi-path routing, fast rerouting, protection switching, faulttolerance and intelligent route management*. Currently, the Internet community employs multi-path routing algorithms like MPLS (Multi-Protocol Label Switching) [181], which is based on managing virtual circuits on top of the IP layer, and hence lacks scalability and robustness. Another approach avoids completely the use of virtual circuits and manages the resources of each session by per-flow fair scheduling of the links. Nevertheless, flows are set up along the shortest paths determined by the underlying routing protocols. The reservation of flows are managed by the Resource Reservation Protocol (RSVP) [297, 260]. However, the deterministic service guarantees are provided to real-time applications using the Interserv architecture [28, 297]. In large networks, this per-flow mechanism does not scale (they can have hundreds of thousands of flows); therefore, in [102], RSVP has been extended by replacing the per-flow routing state with per-source/destination routing state. This results in a state size that grows only quadratically with the number of nodes. Both of these protocols suffer from serious performance bottlenecks because they utilize the single-path routing algorithm *Open Shortest Path First* (OSPF) at the IP layer. Consequently, the bandwidth of the single path is quickly consumed, which results in a high call-blocking probability [260].

The major challenge in *traffic engineering* in a nutshell is *to design multi-path routing protocols for IP networks in which multiple/alternative paths are efficiently discovered and maintained between source and destination pairs*. Such routing protocols will provide solutions to existing technical challenges by using the connectionless paradigm of the IP layer.

## 1.1 Motivation of the Work

We believe that a complete reengineering of the network layer is the logical solution not only to the *traffic engineering* problem but also to network management. The growth of the Internet demands design and development of novel and intelligent routing protocols that result in an intelligent and knowledgeable network layer. Currently, the network layer is relegated to just switching data packets to the next hop based on the information in the routing tables collected by non-intelligent control packets. The new protocols, however, have to be designed with a careful engineering vision in order to reduce their communication, processing, and router's resource costs.

The research in agent-based routing systems has resulted in our developing many novel networking systems [250, 51, 107, 164]. The algorithms utilize software agents which have the following properties [303]:

- *Autonomous:* the capability of performing autonomous actions.
- *Proactive:* the capability of exhibiting opportunistic and goal-oriented behavior and taking initiative where appropriate.
- *Responsiveness:* the capability of perceiving the environment and responding in a timely fashion to the changes that occur in it.
- *Social:* the capability of interaction with other artificial agents and humans when appropriate in order to achieve their own objectives and to help others in their activities.

This design paradigm, therefore, focuses on robust and intelligent agent behavior. In [281], White blames the Artificial Intelligence (AI) community for this. The AI community has been strongly influenced by *Symbol Hypothesis* [176] and first-order predicate logic. The symbols and theorem proving are classical tools, based on the *Resolution Principle* [196]. Consequently, such systems coordinate their activities by exchanging symbolic information and theorem proving. In addition, all properties of a system could not be inferred by representing knowledge in a symbol formula and then manipulating it using first-order predicate logic [204, 281]. Another shortcoming is the *Frame* problem, which results from the need to specify states and state transitions. The measured data obtained from real-world systems has to be represented in symbols, which leads to the *sensor fusion* problem. Connectionist systems and artificial neural networks try to overcome these problems. However, their black box nature makes it difficult to synthesize and utilize them in distributed network systems [281].

The real-world networks represent a dynamic environment in which good routing decisions need to be taken in real time under a number of performance and cost constraints; therefore, applying such complex paradigms to achieve intelligence in the network layer is not feasible. The processing complexity and communication cost of launching such complex agents will be overwhelming, and they will also consume significant amounts of a router's resources, especially in large networks.

The above-mentioned problems in traditional agent-based approaches could be easily solved if we followed a dramatically novel paradigm for designing the agents: *agents need not be rational in order to solve complex problems* [281]. This conjecture, at first, appears to completely boggle the mind because it suggests that *intelligence could result from simple non-intelligent agents*. However, systems based on this design paradigm are rigorously studied in Swarm Intelligence [21]. It takes inspiration from self-organization in natural colony systems, e.g., ants' or bees' [33], and utilizes their principles as a metaphor to design simple agents that take decisions based on local information without the need of a central complex controller. However, such agents are situated in their environment and they utilize either a direct agent-agent communication paradigm or an agent-group paradigm in which they indirectly communicate through the environment. In [33, 24], the authors have defined the basic ingredients of self-organization, which are the following:

1. The *positive feedback* in the system amplifies the good solutions that the agents have discovered. Consequently, other agents are recruited to exploit these good solutions.

2. The *negative feedback* in the system helps in counterbalancing the positive feedback; as a result, good solutions cannot dominate forever.
3. *Amplification of random solutions* helps in discovering and exploring new solutions.
4. *Multiple interactions* help in enabling individuals to use the results of their own activities as well as of others' activities.

In this way a colony is able to achieve a complex and intelligent behavior at the colony level that is well beyond the intelligence and capabilities of an individual in the colony. We believe that self-organization systems have all the features that we could wish for in large network systems.

## 1.2 Problem Statement

We believe that the complexity of the manifold task of endowing intelligence and knowledge to the network layer through self-organizing agents, which are inspired by the communicative and evaluative principles of a honeybee colony, is overwhelmingly phenomenal. Therefore, in our research, we take a cardinal first step to achieve this objective. Our problem statement could be outlined as: *efficient, scalable, robust, fault-tolerant, dynamic, decentralized and distributed solutions to traffic engineering could be provided within the existing connectionless model of IP through a nature-inspired population of agents, which have simple behavior. The agents explore multiple paths between all source/destination pairs and then distribute the network traffic on them. This approach could significantly enhance the network performance*.
Our routing protocol should be able to meet the following challenging requirements:

1. The *agents* must not require existing Multi-Agent System (MAS) software for their realization. Rather, their behavior and learning algorithm should be simple enough to be implemented directly in the network layer by utilizing semantics of C/C++ languages.
2. The processing complexity of *agents* must be kept at a minimum level and the time a router spends in processing them should only be a fraction of the time that it spends in switching data packets. This requirement is necessary because the performance of a router could significantly degrade if *agent processing* steals most of its time [295].
3. The *agents* must explore the network in an asynchronous manner.
4. The protocol must be robust against loss of *agents*.
5. The size of agents must be such that they could fit into the payload of an IP packet. This requirement will significantly reduce communication-related costs.
6. The protocol must be able to scale to large networks.
7. It must be designed with a vision to install it on real-world routers. Therefore, the simulation model must be realizable inside the network stack of a Linux router.
8. It must be realizable in real-world routers without the need for additional resources in both hardware and software. This requirement would simplify its installation, though in a cost-effective manner, on existing routers.

9. It must not require synchronization of clocks in the network.
10. It must not require that the routing tables of different routers should be in a consistent state for taking correct routing decisions.

### 1.2.1 Hypotheses

The study of honeybees has revealed a remarkable sophistication of their communication capabilities. Nobel laureate Karl von Frisch deciphered and structured these into a language in his book *The Dance Language and Orientation of Bees* [259]. Upon their return from a foraging trip, bees communicate the distance, direction, and quality of a flower site to their fellow foragers by waggle dances on a dance floor inside the hive. By dancing zealously for a good foraging site they recruit foragers for it. In this way a good flower site is exploited, and the number of foragers at this site are reinforced. A honeybee colony has many features that are desirable in networks:

- efficient allocation of foraging force to multiple food sources;
- different types of foragers for each commodity;
- foragers evaluate the quality of food sources visited and then recruit the optimum number of foragers for their food source by dancing on a dance floor inside the hive;
- no central control;
- foragers try to optimize the energetic efficiency of nectar collection and make decisions without any global knowledge of the environment.

In our work we use the following hypotheses

(a) **H1:** If a honeybee colony is able to adapt to countless changes inside the hive or outside in the environment through simple individuals without any central control, then an agent system based on similar principles should be able to adapt itself to an ever-changing network environment in a decentralized fashion with the help of simple agents who rely only on local information. This system should be dynamic, simple, efficient, robust, flexible, reliable, and scalable because its natural counterpart has all these features.
(b) **H2:** If designed with a careful engineering vision, nature-inspired solutions are simple enough to be installed on real-world systems. Therefore, their benefit-to-cost ratio should be better than that of existing real-world solutions.

We believe that all of these objectives can be achieved by contemplating novel paradigms for developing agents. The research, however, is of multidisciplinary nature because it involves cross-fertilization of ideas from biology, AI, agent technology, network management, and network engineering. Therefore, we developed a *Natural Engineering* approach[1] to successfully accomplish our objectives in a given time frame.

[1] The focus of our work is on following an engineering approach for nature-inspired routing protocols. However, the engineering approach itself is general enough and complements the existing approaches of Bionik [175, 199] and CI (Computational Intelligence) [3].

## 1.3 An Engineering Approach to Nature-Inspired Routing Protocols

In this section we will introduce our engineering approach[2], which we followed in the design and development of a routing protocol inspired by a natural system (a honeybee colony).

**Definition 1 (Natural Engineering)** *Natural Engineering is an emerging engineering discipline that enables scientists and engineers in search of efficient or optimal solutions for real-world problems under resource constraints to take inspiration and utilize observations from organizational principles of natural systems, and to transform them into structural principles of software organization of algorithms or industrial product design.*

The above-mentioned concept emphasizes six aspects:

1. Understanding the working principles of natural systems.
2. Developing algorithmic models of the organizational principles of natural systems.
3. Understanding the operational environment of target systems.
4. Mapping concepts from the natural system to the technical system.
5. Adapting the algorithmic model to the operational environment of a technical system.
6. Following a testing and evaluating feedback loop in search of optimum solutions under the resource constraints (time, space, computation, money, labor, etc.).

There is no clear-cut way to achieve a perfect match between structures and principles in natural life organizations and working principles in technical systems. The most important challenge, therefore, is to identify a natural system of which the working principles could be appropriately abstracted for deriving suitable principles to work in a given technical system. Instead of adding numerous non-biological features to a natural system, we believe that it is more advisable to look to other natural systems for inspiration. In our case we chose honeybee colonies because the foraging behavior of bees could be transformed into different types of agents performing different routing tasks in telecommunication networks. Both systems have to maximize the amount of a commodity (nectar delivered to hives and data delivered to nodes respectively) as quickly as possible, under a permanently and even unpredictably changing operating environment.

The major focus of research is to design and develop cost-efficient bio/nature-inspired business solutions for highly competitive markets. Therefore, the development of a nature-inspired routing algorithm must follow a feedback-oriented engi-

[2] This section is reproduced by permission of the publisher, Chapman & Hall/CRC Computer and Information Science, from our Chapter 21:*BeeHive: New Ideas for Developing Routing Algorithms Inspired by Honey Bee Behavior* (pages 321–339), published in Handbook of Bioinspired Algorithms and Applications, Albert Zomaya and Stephan Olariu, editors, 2005.

neering approach (see Figure 1.1) that incorporates most of the features discussed above.

First, we considered the ensemble of constraints under which the envisioned routing protocol is supposed to operate:

- Nonavailability of a global clock for trip time calculation.
- Routers and links could crash.
- Routers have limited queue capacity.
- Links have a BER (bit error rate) associated with them.
- The requirements from the Linux kernel routing framework needed to support the protocol.
- The requirements of the IP protocol, which is currently used in the network layer of the Internet.

At the same time we decided that the *bee agents* should explore the network, collect important parameters, and make the routing decisions in a decentralized fashion (in the style in which real scouts/foragers make decisions while collecting nectar from flowers). *Bee agents* should measure the quality of a route and then communicate it to other *bee agents* like foragers do in nature. The structure of the routing tables should provide the functionality of a dance floor for exchanging information among *bee agents* as well as among *bee agents* and data packets. Moreover, we must be able to realize it in a real kernel of the Linux operating system later on.

We implemented our ideas in a simulation environment and then refined our algorithmic mapping through the feedback channel 1 (see Figure 1.1). During this phase we did not use any simulation-specific features that were not available inside the Linux kernel, e.g., vector, stack, or similar data structures. Once we reached a relative optimum of our protocol in a simulator, we started to develop an engineering model of the algorithm. The engineering model can be easily transported to the Linux kernel routing framework. We tested it in the real network of Linux routers and refined our engineering model through the feedback channel 2 (see Figure 1.1). We evaluated our conceptual approach in two prototype projects: *BeeHive* [273], which deals with the design and development of a routing algorithm for fixed networks, and *BeeAdHoc*, the goal of which is to design and develop an energy-efficient routing algorithm for Mobile Ad Hoc Networks (MANETs) [269, 270, 271].

## 1.4 The Scientific Contributions of the Work

In this section we will list the general scientific contributions achieved during our research in the past six years. The reader will appreciate the overwhelming complexity of the work due to the diverse nature of accomplishments achieved in the *BeeHive* and *BeeAdHoc* projects. Some of the information might be duplicated here, but we believe that it is important to make the section self-contained.

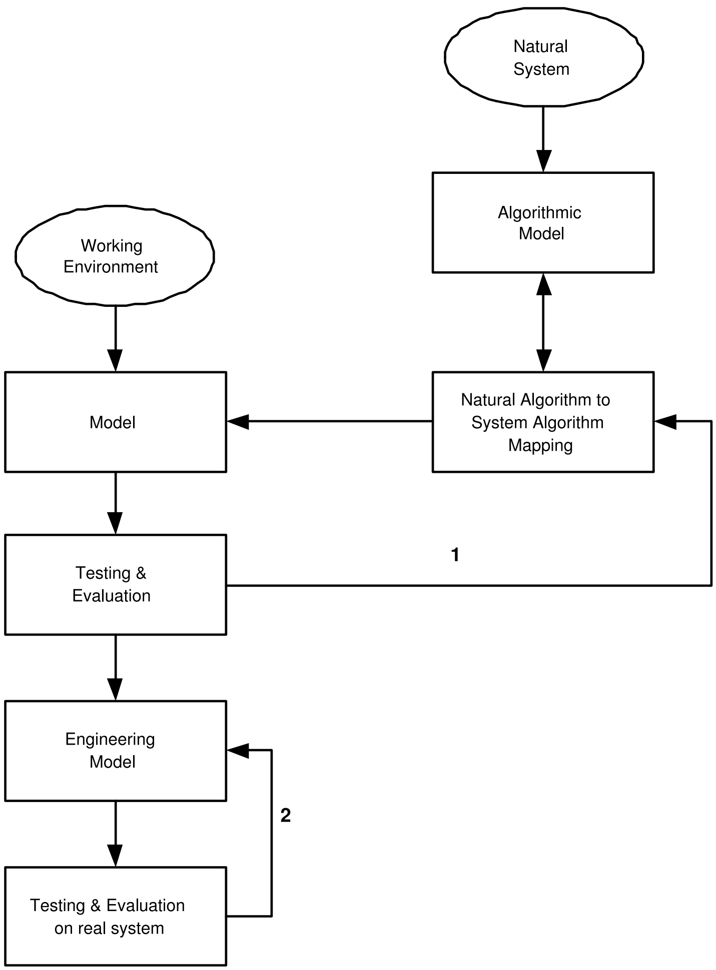


**Fig. 1.1.** Natural protocol engineering

### 1.4.1 A Simple, Distributed, Decentralized Multi-Agent System

We have developed a simple and distributed multi-agent system in which a population of agents collectively achieves an objective. The agents are simple entities with limited processing and memory capabilities and they make their decisions based on their local view of the network state. The state is determined by local information, which is collected in a small region around their launching node. Such a simple agent model is the result of borrowing communication principles from the wisdom of the hive. The agents try to undertake the daunting task of optimizing a number of competing performance values like throughput, packet delay, etc. under different cost constraints.

### 1.4.2 A Comprehensive Routing System

The multi-agent system, as described above, was instrumental in designing and developing a multi-path routing protocol, *BeeHive*, which is dynamic, simple, efficient, robust, flexible, and scalable. As demonstrated by our results, the algorithm achieves a similar or better performance than the existing state-of-the-art algorithms. *BeeHive*, however, achieves this objective with significantly lesser costs in terms of processing, communication, and router's resources. The algorithm does not require access to the complete network topology; rather, it works with a local view of the network. The agents take their decisions in an autonomous and decentralized fashion.

### 1.4.3 An Empirical Comprehensive Performance Evaluation Framework

The other major contribution of the work is a comprehensive performance evaluation framework, which calculates a number of important performance values and the associated costs of a routing algorithm. The framework can also vary a number of network configurations from traffic patterns to network topology. As a result, the developer of a routing protocol can study the behavior of an algorithm on a wide operational landscape with a focus on its benefit-to-cost ratio in an unbiased manner. The framework proved to be useful in identifying reasons behind the anomalous behavior of *BeeHive* in different scenarios. Subsequently, we were able to improve our algorithm through the feedback channel 1 as shown in Figure 1.1.

### 1.4.4 A Scalability Framework for (Nature-Inspired) Agent-Based Routing Protocols

We developed a comprehensive framework that facilitates the study of the scalability of agent-based distributed systems in general and of routing protocols in particular. The framework provides a formal model and a set of empirical tools to protocol developers that are useful in investigating the scalability of their protocols at an early stage of development. To our knowledge, this is the first model that provides an unbiased way of studying the scalability of (nature-inspired) agent-based routing protocols.

### 1.4.5 Protocol Engineering of Nature-Inspired Routing Protocols

One of the most important contributions of our work is the vision of *Natural Engineering*, introduced in the last section. We believe that developing a nature-inspired system, which can be installed or utilized in real-world systems, is a challenging task. The nature-inspired community, at times, lacks vision about the real operational environments. As a result, most of the proposed solutions are never realized in the intended real-world systems. Our work, according to our knowledge, is an important step from "Swarm Intelligence" to "Natural Engineering." We believe that the work will stimulate other researchers to adopt a similar approach for their projects as well.

### 1.4.6 A Nature-Inspired Linux Router

Our *Natural Engineering* approach significantly helped us in developing an algorithmic model in the simulation environment that is mostly independent of the underlying features of a simulation system. It rather utilizes only those components in a simulation environment which are available in real-world Linux routers. This approach showed its benefits once we started developing an engineering model in the form of a nature-inspired Linux router because we were able to make a quantum leap with significantly limited man power and computing resources.

### 1.4.7 The Protocol Validation Framework

Another important contribution of the work is a comprehensive validation framework in which we implemented the same traffic generators in the simulation and in an application layer of a Linux network stack. We also utilized the same network topology in both simulation and the real network of Linux routers. Our validation principle is: if we generate the same traffic patterns in identical topologies in both simulation and the real network, then the performance values of the algorithms should be traceable from one environment to another with acceptable deviations. To the best of our knowledge, *BeeHive* is the first nature-inspired algorithm that has been implemented in real networks and shown substantial performance benefits for existing real-world applications.

### 1.4.8 The Formal Framework for Nature-Inspired Protocols

An important contribution of our research is a formal framework developed by utilizing probabilistic recursive functions and formal concepts of M/M/1 queuing theory. By utilizing the model, we were able to model both agents' and data traffic flows passing through a node in a network running our *BeeHive* protocol. We then used this traffic flow model to formally represent relevant performance parameters. We believe that the framework is generic and will help protocol designers and developers to model the behavior of their nature-inspired routing protocols by utilizing its relevant concepts. In line with our *Natural Engineering* approach we validated our formal model by comparing its estimated values with the values obtained from the OMNeT++ network simulator. The performance metrics estimated by the formal model approximately map to the metrics obtained from the simulator.

### 1.4.9 A Simple, Efficient, and Scalable Nature-Inspired Security Framework

Another important contribution of the work is the conducting of a pilot study for the vulnerabilities of our *BeeHive* protocol that malicious nodes can exploit to disrupt normal network operations. To the best of our knowledge, this is the first detailed pilot study within nature-inspired community. Consequently we developed an immune-inspired simple, efficient, and scalable nature-inspired security framework for *BeeHive* that provides the same security level as that of signature-based cryptographic solutions but at significantly smaller processing and communication costs. Our results show that our enhanced framework can counter a number of threats launched by malicious nodes in the network.

### 1.4.10 Emerging Mobile and Wireless Sensors Ad Hoc Networks

Another important contribution of our work is demonstrating that bee-inspired protocol engineering is not limited to just fixed telecommunication networks. We show that by taking inspiration from the wisdom of the hive, we can also develop an energy-efficient routing protocol, *BeeAdHoc*, for Mobile Ad Hoc Networks (MANETs), and *BeeSensor*, for Wireless Sensor Networks (WSNs). Both protocols take inspiration from the energy conservation behavior of a bee colony. Following our *Natural Engineering* approach, we implemented *BeeAdHoc* on mobile laptops running Linux and tested our protocol in a real-world MANET. We also designed a novel testing methodology in which we gradually move from a simulator-only environment to real MANET. This work shows the potential of nature-inspired protocols in next-generation networks.

## 1.5 Organization of the Book

The work presented in this book is organized into nine chapters. Each chapter, except the first and the last, will provide a comprehensive review of the research conducted in a particular phase of our *Natural Engineering* cycle, from our conceiving the ideas from the working principles of a natural system, to our developing an algorithmic model from them, to our realizing the algorithmic model both in a simulation environment and in a real network of Linux routers. The realization phase, both in simulation and real networks, is complemented by extensive testing, analysis, evaluation, and feedback channels.

### Chapter 2: A Comprehensive Survey of Nature-inspired Routing Protocols

The chapter presents the true challenges that a routing protocol is expected to meet in complex networks of the new millennium. We provide classifications of the algorithms based either on their characteristics or on their design philosophy. The basic objective of the survey is to understand the design doctrine of different communities involved in the design and development of routing algorithms. This will motivate researchers to develop state-of-the-art routing algorithms through a process of cross-fertilization of useful features and characteristics of different design doctrines. We classify the communities into three categories: Networking, Artificial Intelligence (AI), and Natural Computing (NC). The focus of the survey presented in Chapter 2 is on the algorithms developed by the Natural Computing community. We provide a detailed survey of routing algorithms inspired from the pheromone-laying principles of ant colonies. The algorithms are based on the Ant Colony Optimization (ACO) metaheuristic. We also provide a comprehensive review of routing algorithms based on the principle of evolution in natural systems. Later in the chapter, we introduce routing algorithms based on the principles of *Reinforcement Learning*. These routing algorithms are developed by the Artificial Intelligence community. Finally, we briefly summarize the routing algorithms recently developed by the networking community. The comprehensive survey proved helpful in identifying the merits and deficiencies of existing state-of-the-art routing protocols developed by different communities. Most of the chapter has been reproduced from the following paper with the kind permission of Elsevier.

- H. F. Wedde and M. Farooq. A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks. *Journal of System Architecture*, 52(8-9):461-484, 2006.

### Chapter 3: From *the Wisdom of the Hive* to Routing in Telecommunication Networks

The chapter describes the most important steps in our *Natural Engineering* approach. It starts with a brief introduction to the foraging principles of a honeybee colony. We present the biological concepts in such a manner that the reader conveniently conceives a honeybee colony as a population-based multi-agent system, in which simple agents coordinate their activities to solve the complex problem of the allocation of labor to multiple forage sites in dynamic environments. The agents achieve this objective in a decentralized fashion with the help of local information they acquire while foraging. We argue that an efficient, reliable, adaptive, and fault-tolerant routing algorithm has to also deal with similar daunting issues.

We then provide the mapping of concepts from a natural honeybee colony to an artificial multi-agent system, which can be utilized for routing in telecommunication networks. The mapping of concepts appears to be a crucial step in developing an algorithmic model of an agent-based routing system. We emphasize the motivation behind important design principles of our *BeeHive* routing algorithm. We provide a comprehensive description of our *bee agent* model by emphasizing the communication paradigm utilized by the *bee agents*, which is instrumental in reducing the costs associated with a routing algorithm: communication, processing, and router's resources. Later in the chapter, we introduce our comprehensive empirical performance evaluation framework that calculates a number of preliminary and auxiliary performance values. These values provide an in depth insight into the behavior of a routing algorithm under a variety of challenging network configurations.

Finally, we introduce our extensive experimental framework in a simulation environment. The experiments are designed through extensive brainstorming exercises in order to meticulously analyze the behavior of a routing protocol under diversified network operations. The results obtained from our performance evaluation framework are discussed. We compare *BeeHive* with a state-of-the-art ACO routing algorithm, *AntNet*, a state-of-the-art evolutionary routing algorithm *Distributed Genetic Algorithm (DGA)*, *OSPF*, and *Daemon*. *Daemon* is an ideal algorithm that can instantly access the complete network topology and size of the queues in all routers to make an optimum routing decision. The algorithm, though, is not realizable in real networks due to the associated costs; but, nevertheless, it serves as an important benchmark for different algorithms.

The results of the experiments unequivocally suggest that *BeeHive* is able to achieve similar or better performance under congested loads compared with *AntNet* and is able to achieve similar or better performance under normal static loads as compared with *OSPF*. However, this excellent performance of *BeeHive* is achieved with significantly smaller communication and processing costs, and routing tables, which have the order of the size as in *OSPF*. The chapter contains extracts from our following published papers, reproduced [273, 268] with the kind permission of Springer Verlag and Chapman & Hall/CRC Computer and Information Science:

1. Horst F. Wedde, Muddassar Farooq, and Yue Zhang. BeeHive: An Efficient Fault Tolerant Routing Algorithm under High Loads Inspired by Honey Bee Behavior. In Marco Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Stützle, editors, Proceedings of the Fourth International Workshop on Ant Colony and Swarm Intelligence (ANTS 2004), volume 3172 of Lecture Notes in Computer Science, pages 83–94, Brussels, Belgium, September 2004. Springer Verlag. (Winner of the Best Paper Award ANTS 2004.)
2. Horst F. Wedde and Muddassar Farooq. A Performance Evaluation Framework for Nature Inspired Routing Algorithms. In Franz Rothlauf et al., editors, Applications of Evolutionary Computing – Proceedings of EvoWorkshops 2005, volume 3449 of Lecture Notes in Computer Science, pages 136–146, Lausanne, Switzerland, March/April 2005.
3. Horst F. Wedde and Muddassar Farooq. *BeeHive: Routing Algorithms Inspired by Honey Bee Behavior*. Künstliche Intelligenz. Schwerpunkt: Swarm Intelligence, pages 18–24, Nov 2005.
4. Horst F. Wedde and Muddassar Farooq. *BeeHive: New Ideas for Developing Routing Algorithms Inspired by Honey Bee Behavior.* In Handbook of Bioinspired Algorithms and Applications, Albert Zomaya and Stephan Olariu, Ed. Chapman & Hall/CRC Computer and Information Science, Chapter 21, pages 321–339, 2005.

**Chapter 4: A Scalability Framework for Nature-inspired Routing Algorithms**

The chapter presents a new scalability framework that designers and developers of the routing algorithms in general, and of nature-inspired routing protocols in particular, can utilize to analyze the scalability of their routing protocols. We believe that our new framework will enable the designers to establish the scalability of their routing protocols in an early stage of *protocol engineering* [140]. Such a framework will be instrumental in practicing the principles of *Software Performance Engineering* (SPE), which also emphasizes the consideration of performance and scalability issues early in the design and architectural phase in order to rectify the deficiencies in a simulation environment. This will not only obviate the risk of a disaster once the algorithm is deployed on large-scale networks, but also avert the cost overruns due to tuning or redesign of the algorithm later in the protocol engineering cycle. Consequently, such a pragmatic protocol engineering cycle will be capable of reducing the time to market of a new protocol.

Our scalability model defines power and productivity metrics for a routing protocol. The productivity metric provides insight into the benefit-to-cost ratio of a routing protocol. The cost model includes the communication, processing, and memory costs related to a routing algorithm. We believe that the productivity of a routing algorithm is an important performance value which can be used for an unbiased investigation of a routing protocol. Later we define a scalability metric, which is a ratio of productivity values of two network configurations, and its value should be ideally 1 if the algorithm is perfectly scalable from one network configuration to the other.

The framework is general enough to act as a guideline for analyzing the scalability of any agent-based network system. However, in our work, we restricted our analysis to only three protocols due to lack of high performance simulation platforms. We studied the scalability behavior of *BeeHive*, *AntNet*, and *OSPF* in six topologies which vary in their degree of complexity and connectivity. The size of the topologies is gradually increased from eight nodes to 1,050 nodes. According to our knowledge, this is the first extensive effort to empirically study the scalability of nature-inspired routing protocols.

The results demonstrate that *BeeHive* is able to deliver superior performance under both high and low network traffic loads in all topologies. We believe that an engineering vision during the design and development phase, in which we emphasized the scalability as an important metric, has significantly helped *BeeHive* in achieving better scalability metrics for the majority of the network configurations compared with *AntNet* and *OSPF*. It took more than six months to extensively evaluate the algorithms under a variety of network configurations.

### Chapter 5: *BeeHive* in Real Networks of Linux Routers

This chapter describes the second phase of our *Natural Engineering* approach: the realization of an engineering model of *BeeHive* inside the network stack of the Linux kernel, and then the comparison of its performance values with *OSPF* in a real network of eight Linux routers. The work presented in the chapter is novel in the sense that, to our knowledge, *BeeHive* is the first nature-inspired routing algorithm which has been realized and tested in real networks.

The chapter begins by illustrating different design options that are available for realizing a nature-inspired routing algorithm in a Linux router. We then describe the motivation behind our *engineering model* that we realized in a Linux router. Subsequently, we define the software architecture of our Nature-inspired Linux router. Here, we emphasize the challenges we encountered because of the unique features of the *BeeHive* algorithm.

We also migrated our *performance evaluation framework* to the application level of the Linux network stack. The motivation behind this significant step is to follow the protocol verification principle: *if we generate the same traffic patterns through the same traffic generators in both simulation and real networks and utilize the same* *performance evaluation framework in both simulation and real networks then the performance values obtained from the simulation environment should be traceable to the ones obtained from the real Linux network with minor deviations, provided our simulation environment depicts a somewhat realistic picture of a real network.* We believe that this verification principle will help in tracking the performance values in simulation with those of their counterparts in real networks. If the values are similar, then this would strengthen our thesis: *Nature-inspired routing protocols, if engineered properly, could manifest their merits in real networks.*

Finally, we discuss the results obtained from extensive experiments both in simulation and in a real network. We feel satisfied because the performance values obtained from the simulation are consistent with the values in the real network, with an acceptable degree of deviation. This, according to our knowledge, is the first substantive work which shows the benefits of utilizing nature-inspired routing protocols in real networks running real-world applications, e.g., File Transfer Protocol (FTP) and Voice over IP (VoIP). The success in this phase satisfyingly concludes our last phase in the protocol development cycle of our *Natural Engineering* approach.

### Chapter 6: A Formal Framework for Analyzing the Behavior of *BeeHive*

In this chapter we report our formal framework to analyze the behavior of our *BeeHive* protocol. The motivation for such a formal framework comes from the fact that most researchers in Natural Computing follow a well-known protocol engineering philosophy: inspire, abstract, design, develop, and validate. Consequently, researchers even today have little understanding of the reasons behind the superior performance of nature-inspired routing protocols. We argue that formal understanding about the merits of *BeeHive* is important in order to get an in depth understanding about its behavior.

We revisit in this chapter our *BeeHive* protocol, which is introduced in Chapter 3, to understand the merits of different design options with the help of our formal framework. We show why different quality formulas provide the same performance. We used probabilistic recursive functions for analyzing online the stochastic packet-switching behavior of the algorithm. The queuing delays experienced due to the congestion have been analyzed using the formal concepts of M/M/1 queuing theory. With the help of this framework we model bee traffic and data traffic on the links of a given node. Using this traffic model we derive formulas for throughput (bits correctly delivered at the destination in unit time) and end-to-end delay of a packet.

Towards the end of the chapter we describe our empirical verification framework in OMNeT++ to validate the correctness of our formal model. We validated our formal model on two topologies and compared its results with the results obtained from the simulations. The estimated performance values of the model have only a small deviation from the real values measured in the network simulator. We believe that this formal treatment will add an important phase of *formal modeling* to protocol engineering of nature-inspired protocols. This formal treatment is the key to widespread acceptability of such protocols in the networking community. The chapter has been reproduced from our following paper with the kind permission of IEEE.

- S. Zahid, M. Shehzad, S. Usman Ali, and M. Farooq. A comprehensive formal framework for analyzing the behavior of nature inspired routing protocols. In Proceedings of Congress on Evolutionary Computing (CEC), pages 180–187. IEEE, Singapore, September 2007.

## Chapter 7: An Efficient Nature-Inspired Security Framework for *BeeHive*

In this chapter we investigate the vulnerabilities and related security threats that malicious nodes in a network can exploit to seriously disrupt the networking operations. Remember that researchers working in nature-inspired protocols always implicity trust the identity and routing information of *ant or bee agents*. This assumption is no more valid in real-world networks where compromised nodes can wreak havoc by launching malicious agents that can significantly alter the routing behavior of a protocol. The lack of any work in this important domain motivated us to undertake research to develop a simple, scalable, and efficient security framework for our *BeeHive* protocol.

We first provide a list of attacks that malicious nodes can launch on a network running the *BeeHive* protocol. We then introduce our *BeeHiveGuard* security framework, a signature-based security framework in which *bee agents* are protected by the use of the principles of Public Key Infrastructure (PKI) against tampering of their identity or routing information. An obvious disadvantage of this approach is that the size of *bee agents* increases manifold because the signatures are added to their payload. Moreover, complex decryption and encryption operations need to be performed at each intermediate node, which increases the processing complexity manifold. Our results indicate that the processing complexity of *bee agents* in *BeeHiveGuard* increased by more than 52,000% and the communication-related costs increased by more than 200% compared to *BeeHive*. Remember that *bee agents* are launched after every second; therefore, this overhead is definitely not acceptable. As a result, we have to look for other design paradigms that provide the same security level as *BeeHiveGuard* but with significantly smaller processing and communication costs.

After initial investigations, Artificial Immune Systems (AISs), inspired by immunology principles, provide a suitable framework for a simple, efficient, and scalable security framework. Our proposed framework, *BeeHiveAIS*, works in three phases: (1) In the learning phase it passively monitors the network traffic to learn the normal traffic patterns; (2) in the second phase it generates a set of detectors that are later used in the protection phase to classify agents that perform suspicious activities as malicious agents; and (3) during the protection phase it protects the system against malicious attacks. *BeeHiveAIS* has a simple anomaly detection algorithm that works without the need to transmit redundant information in the *bee agents*. Consequently, it has significantly smaller processing and communication costs compared to *BeeHiveGuard*.

We developed an empirical validation framework to verify that both frameworks are able to successfully counter a number of attacks launched by malicious nodes.

We tested both frameworks on topologies ranging from four to 150 nodes. The conclusion of the experiments is: BeeHiveAIS *provides the same security level as* BeeHiveGuard *does, but with significantly smaller processing and communication costs. Moreover, the relevant performance values of the* BeeHiveAIS *are within an acceptable range of those of* BeeHive *(without any attack).* The chapter contains extracts from our following published papers, reproduced with the kind permission of Springer Verlag.

1. H. F. Wedde, C. Timm, and M. Farooq. BeeHiveGuard: A step towards secure nature inspired routing algorithms. In Applications of Evolutionary Computing, volume 3907 of Lecture Notes in Computer Science, pages 243–254. Springer Verlag, April 2006.
2. H. F. Wedde, C. Timm, and M. Farooq. BeeHiveAIS: A simple, efficient, scalable and secure routing framework inspired by artificial immune systems. In Proceedings of the PPSN IX, volume 4193 of Lecture Notes in Computer Science, pages 623–632. Springer Verlag, September 2006.

**Chapter 8: Bee-Inspired Routing Protocols for Mobile Ad Hoc and Sensor Networks**

Towards the end of our book, we highlight the potential of nature-inspired routing protocols for emerging networks like Mobile Ad Hoc Networks (MANETs) and Wireless Sensor Networks (WSNs). Both types of networks are becoming popular because of their potential utility in war theaters, disaster management, security and tactical surveillance, and weather monitoring. The typical characteristic of these networks are that they are deployed in the real world without any requirement for an infrastructure. As a result, each node is delegated the task of routing as well. Due to mobility in MANETs and varying power levels in WSNs, the connectivity of nodes continuously keep on changing. This calls for energy-efficient power-aware self-organizing routing protocols.

We introduce our *BeeAdHoc* routing protocol for MANETs that delivers the same or better performance compared to existing state-of-the-art MANET routing protocols. But its energy consumption is significantly smaller than that of other protocols. Following our *Natural Engineering* approach we also implemented *BeeAdHoc* in Linux on mobile laptops. We developed a novel real-world testing strategy that gradually moves the testing environment from simulation to real MANETs. The results of our experiments indicate the same pattern: *BeeAdHoc delivered the same or better performance as compared to state-of-the-art algorithms but at significantly smaller communication costs.*

Finally, we describe our *BeeSensor* protocol for WSNs. *BeeSensor* tries to combine energy efficiency of *BeeAdHoc* with the scalability properties of *BeeHive*. We compared the *BeeSensor* with state-of-the-art nature-inspired routing protocols in a real target tracking application. The results indicate that *BeeSensor* delivers the same or better performance compared to existing algorithms but has significantly smaller processing and communications costs. The results reported in the chapter are intriguing enough to motivate researchers to develop self-organizing, simple, scalable, adaptive, and efficient routing protocols for emerging next-generation networks. The chapter contains extracts from our following published papers, reproduced with the kind permission of ACM, IEEE and Springer Verlag:

- H. F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat. BeeAdHoc: an energy efficient routing algorithm for mobile ad-hoc networks inspired by bee behavior. In Proceedings of GECCO, pages 153–161, Washington, June 2005.
- H. F. Wedde and M. Farooq. The wisdom of the hive applied to mobile ad-hoc networks. In Proceedings of the IEEE Swarm Intelligence Symposium (SIS), pages 341–348, Pasadena, June 2005.
- M. Saleem and M. Farooq. BeeSensor: A Bee-inspired power aware routing protocol for wireless sensor networks. In M. Giacobini et al. (Eds.), volume 4449 of Lecture Notes in Computer Science, pages 81–90. Springer Verlag, April 2007. 109.
- M. Saleem and M. Farooq. A framework for empirical evaluation of nature inspired routing protocols for wireless sensor networks. In Proceedings of Congress on Evolutionary Computing (CEC), pages 751–758. IEEE, Singapore, September 2007.

### Chapter 9: Conclusion and Future Work

In this chapter, we summarize the contributions of our work. We stress the need for the *Natural Engineering* approach because this significantly helped us in successfully designing a dynamic, simple, efficient, robust, flexible, and scalable multi-path routing algorithm and then installing it in a real network of Linux routers. We believe that a similar approach can help in realizing other nature-inspired algorithms in their respective real environments.

We conclude the chapter with interesting future directions. The most important one is: design and development of a dedicated nature-inspired router in hardware which optimally runs nature-inspired routing algorithms. Before this step is taken, we have to reengineer *BeeHive* in such a fashion that it is capable of seamlessly replacing *OSPF* in the existing packet-switched IP networks.

Springer

http://www.springer.com/978-3-540-85953-6

Bee-Inspired Protocol Engineering
From Nature to Networks
Farooq, M.
2009, XX, 306 p., Hardcover
ISBN: 978-3-540-85953-6