NATURE-INSPIRED CONGESTION CONTROL AND AVOIDANCE IN WIRELESS SENSOR NETWORKS

Pavlos Antoniou

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Cyprus

Recommended for Acceptance

by the Department of Computer Science

May, 2012

APPROVAL PAGE

Doctor of Philosophy Dissertation

NATURE-INSPIRED CONGESTION CONTROL AND AVOIDANCE IN WIRELESS SENSOR NETWORKS

Presented by

Pavlos Antoniou

Research Supervisor _____
Prof. Andreas Pitsillides

Committee Member _____
Prof. Chris Christodoulou

Committee Member _____
Prof. Vasos Vassiliou

Committee Member _____
Prof. Marios Polycarpou

Committee Member _____
Prof. Giovanna Di Marzo Serugendo

University of Cyprus

May, 2012

NATURE-INSPIRED CONGESTION CONTROL AND AVOIDANCE IN WIRELESS SENSOR NETWORKS

Pavlos Antoniou

University of Cyprus, 2012

Complex collective behaviors and collective intelligence in groups of animals and insects found in nature appear appealing for Wireless Sensor Network (WSN) congestion control because they emerge from simple local interactions of the individuals in the system. Animal and insect societies possess inherent properties like simplicity (e.g. operate on the basis of a few simple rules), scalability (e.g. controllable behavior as the size of the system scales up), self-adaptiveness to internal or external changes, and robustness against threats (e.g. system failures).

These properties are desirable in a WSN environment, whose constituent node components can be constrained in terms of computation capability, memory space, communication bandwidth and energy supply. Effective and efficient congestion control approaches are expected to be self-adaptive to the dynamically changing network and traffic conditions, robust against failures, and scalable as the size of the network increases.

The thesis objective is to investigate nature inspired techniques in the context of congestion control in WSNs. In particular, properties such as simplicity at the individual node level, emerging behavior at the global network level and intrinsic

properties of selected nature inspired techniques such as robustness, self-adaptation, and scalability are evaluated. Specifically, this thesis draws inspiration from swarm intelligence (SI) and mathematical models of population biology.

From the viewpoint of swarm intelligence, a considerable number of models based on self-propelled particles have been developed to solve a variety of problems by means of collective motion. The research framework behind the majority of network-oriented studies involving self-propelled particles was fueled by the ACO theory proposed by Dorigo et al. This theory was successfully involved in network-oriented studies, especially in the field of ad-hoc and mobile ad-hoc networks (MANETs). Previous work on congestion control involving mathematical models of population biology is basically applicable to the Internet on the basis of either improving the current TCP congestion control mechanism or providing a new way of thinking and combating congestion.

This thesis proposes two novel nature-inspired congestion control and avoidance approaches that aim at improving performance, in particular with respect to packet delivery ratio, end-to-end delay and energy consumption in a variety of sensor network applications that follow either the event-based or the continuous-based (streaming) data delivery model. Both proposed congestion control approaches embody a nature-inspired development drive involving the aforementioned desirable properties. These properties are not explicitly programmed into the network, but

emerge as a result of the collective behavior among interacting individuals. In addition, the proposed approaches are simple to implement at the individual node, involve minimal information exchange, and provide graceful performance degradation at low, medium and high traffic loads.

The first approach targets event-based applications and adopts a swarm intelligence paradigm inspired by the obstacle avoidance behavior and the orientation behavior of bird flocks having global self-* properties (e.g. self-adaptation) achieved collectively without explicitly programming them into individual nodes. The main idea is to 'guide' packets (birds) to form flocks and flow towards the sink node (global attractor), whilst trying to avoid congestion regions (obstacles). In the proposed flock-based congestion control (Flock-CC) approach, the motion of each packet is influenced by: (a) the repulsive and attractive forces among closely located individuals, (b) the limited visual perception that defines the field of view (FoV), (c) the artificial magnetic field towards a global attractor, and (d) randomness. Performance evaluations show the effectiveness of the Flock-CC approach in balancing the offered load by exploiting available network resources. Flock-CC provides graceful performance degradation in terms of packet delivery ratio, packet loss, delay and energy tax as the traffic load increases to even extreme levels. In addition, the proposed approach achieves adaptation to changing network and traffic conditions, robustness against failing nodes, even at extreme cases, scalability to different network sizes, and is shown to outperform typical conventional approaches.

The second approach targets streaming applications and focuses on how congestion can be prevented, or if not gracefully controlled, in small-scale networks by regulating the rate of each traffic flow based on the Lotka-Volterra population model. The Lotka-Volterra based congestion control (LVCC) strategy involves minimal exchange of information and computation burden and is simple to implement at the individual node. Performance evaluations reveal that the LVCC approach achieves adaptability to changing traffic loads, scalability and fairness among flows, while providing graceful performance degradation, in terms of throughput and delay of individual streams, as the offered load increases. However, its scalability is questionable, and further work is required here, as for example the adaptive setting of its control parameters.

This thesis successfully adapts techniques from nature and demonstrates their usefulness in combating congestion in wireless sensor networks under changing network and traffic conditions in the sense of reducing packet losses and thus retransmissions, leading to decreased delay in moving packets to the sink and low levels of energy expenditure.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed in the preparation and completion of this study.

First and foremost, I would like to express my utmost gratitude to Dr. Andreas Pitsillides, Professor at the Department of Computer Science of the University of Cyprus, whose sincerity and encouragement I will never forget. Heartfelt thanks go to him who has given me so much energy for research study and meaningful advice on nearly every page of this work. I'm thankful to him for the numerous discussions that I have had with him and for the suggestions that I received from him. His careful reading of this thesis improved the content and its form.

This work would not have been realized without enlightening discussions and a fruitful collaboration with Dr. Tim Blackwell, Dr. Andries Engelbreht and Dr. Loizos Michael. I am grateful for the opportunity to thank them all here.

I would like to express my special appreciation and thanks to Dr. Tim Blackwell, Senior Lecturer at the Department of Computing in Goldsmiths College of the University of London, who has been a real inspiration for me towards introducing an innovative nature-inspired approach to the area of networks. With his expertise

in the field of swarm intelligence he extended my knowledge beyond the concepts of conventional computer science approaches.

I am heartily grateful to Dr. Andries Engelbreht, Professor at the Department of Computer Science of the University of Pretoria, South Africa, for his countless advice and for providing lots of good ideas. His expertise in the field of computational intelligence was quite valuable in developing the proposed approaches. I am grateful to him for reading the majority of my papers carefully at a very busy time and providing valuable suggestions and corrections in order to take out the best results.

Furthermore, I would like to express my gratitude to Dr. Loizos Michael, Lecturer at the Open University of Cyprus. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped to clarify and highlight different aspects of the proposed nature-inspired approaches. His support, unfailing patience, and unselfish attitude were much appreciated.

I would like to thank Dr. Chris Christodoulou and Dr. Vasos Vassiliou both members of the faculty of the Computer Science Department of the University of Cyprus for serving as readers of my thesis and for providing valuable comments.

I would like to acknowledge the academic and technical support of the Department of Computer Science and especially the Networks Research Lab (NetRL) which have provided the support and equipment I have needed to produce and complete my thesis. I am indebted to all student colleagues, members of NetRL, for providing a stimulating and fun environment in which to grow and learn.

This work was made possible by the financial support of the "GINSENG: Performance Control in Wireless Sensor Networks" project funded by the 7th Framework Programme under Grant No. ICT-224282 and the "MiND2C: Mimicking Nature for Designing Robust Congestion Control Mechanisms in Self-Organized Autonomous Decentralized Networks" project funded by the Research Promotion Foundation of Cyprus under Grant No. TPE/EPIKOI/ 0308(BE)/03.

I wish to express a very special thanks to my family for providing a loving and caring environment for me from the very beginning of this study till the end. My brother, Zinonas, and my sister, Anna, were particularly supportive, but especially my parents who bore me, raised me, supported me, taught me, and loved me.

Lastly, and most importantly, I wish to thank my beloved wife, Christina, for her patience and love during the long journey to Ithaca. Her support and untiring efforts behind the scene everyday have sustained me through many difficulties and made everything possible. To her, I dedicate this thesis with deepest gratitude.

I could not forget The One above all of us, The omnipresent God, for answering my prayers, for giving me the courage and the strength to deal with all difficulties, challenges and problems in order to get through my studies successfully. I thank our Lord Jesus Christ from the bottom of my heart for all of His blessings in every aspect of my life.

«Πᾶσα δόσις ἀγαθὴ καὶ πᾶν δώρημα τέλειον ἄνωθέν ἐστι, καταβαῖνον ἐκ σοῦ τοῦ Πατρὸς τῶν φώτων», Θεία Λειτουργία Αγίου Ιωάννου του Χρυσοστόμου.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acronyms

ACO Ant Colony Optimization

AIMD Additive Increase Multiplicative Decrease

ANSI Ad-hoc Networking with Swarm Intelligence

AntHocNet

AntSensNet

AODV Ad-hoc On-demand Distance Vector

AQM Active Queue Management

BGR Biased Geographical Routing

BICC Bio-Inspired Congestion Control

CAR Congestion Aware Routing

CC Congestion Control

CODA Congestion Detection and Avoidance

CSMA Carrier Sense Multiple Access

EED End to End Delay

FlockCC Flock-based Congestion Control

FoV Field of View

GPS Global Positioning System

LV Lotka Volterra

LVCC Lotka-Volterra based Congestion Control

MAC Medium Access Control

MANET Mobile Ad-hoc Network

PDA Personal Digital Assistant

PDR Packet Delivery Ratio

PSO Particle Swarm Optimization

QoS Quality of Service

SI Swarm Intelligence

TADR Traffic-Aware Dynamic Routing

TCP Transport Control Protocol

TDMA Time Division Multiple Access

WSN Wireless Sensor Network

ZoA Zone of Attraction

ZoO Zone of Orientation

ZoR Zone of Repulsion

Chapter 1

Introduction

1.1 The Problem

Rapid technological advances and innovations in the area of autonomous systems push the vision of Ambient Intelligence from concept to reality. Towards this direction, there has been an unprecedented research interest for autonomous networked systems with emphasis on WSNs [16]. WSNs are deployed for several mission-critical tasks (e.g. as platforms for health monitoring, process control, environmental observation, battlefield surveillance) and are expected to operate unattended (without human intervention) for extended periods of time.

Typically, WSNs comprise of small (and often cheap), cooperative devices (nodes) which may be (severely) constrained in terms of computation capability, memory space, communication bandwidth and energy supply. Increasingly, with the rapid development of low-cost hardware CMOS cameras and microphones, autonomous

sensor devices are becoming capable of ubiquitously retrieving multimedia content such as audio and low-rate video streams from the environment [17].

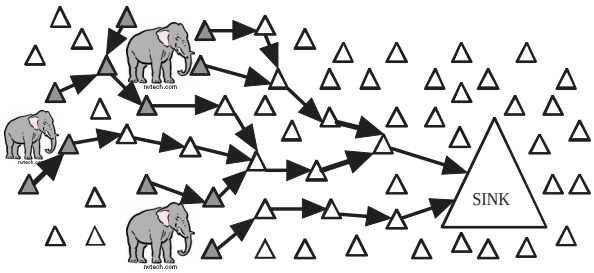


Figure 1: A WSN for wildlife monitoring.

In the context of WSN, autonomous nodes may interact (a) with the environment so as to sense or control physical parameters, and (b) with each other in order to exchange information or forward data towards one or more dedicated sink nodes. Typically, WSNs operate under light load, but large, sudden, and correlated-synchronized impulses of data may suddenly arise in response to a detected or constantly monitored event. More particularly, a number of nodes that at a particular moment sense an event (as for example the grey-shaded nodes of Fig. 1), will generate data and inject them into the network. Alternatively, some nodes may be constantly generating streaming data. All the amount of data must be directed in a multi-hop manner to the sink node(s).

Large numbers of generated packets in conjunction with variable wireless network conditions, may result in unpredictable behavior in terms of traffic load variations and link capacity fluctuations. The problem is worsened due to topology changes driven by node failures, mobility, or intentional misbehavior. Under these stressful

situations which are likely to occur in WSN environments and are expected to provoke congestion.

Congestion conditions occur quite often, when the traffic load injected into the network exceeds available capacity at any point of the network. The problem of congestion in wireless sensor networks is unveiled in the beginning of this study, while it is thoroughly investigated and effectively tackled in the remainder of this study.

Congestion control involves measures taken for controlling the traffic injected into the network in order to avoid or mitigate congestion collapse [18]. Congestion control is considered to be one of the most basic components of a performance controlled network. Robust, scalable and self-adaptable congestion control approaches aim to keep the network operational under congestion conditions, whilst keeping packet loss and end-to-end delay within tolerable levels as well as maximizing network lifetime.

1.2 Motivation

Early studies in the area of wireless sensor networks had mainly focused on more fundamental networking problems, e.g. medium access control (MAC), topology, routing, and energy efficiency, targeting applications in which network performance assurances are not considered essential, such as agriculture and environmental monitoring. Lately, with the emergence of mission-critical applications (e.g. health monitoring, plant automation), there has been increased interest [19] in providing

performance assurances, especially for metrics such as packet delivery ratio, delay and energy consumption.

Increasingly, in the near future, various applications based on multimedia wireless networks are expected to arise, where many types of sensors such as cameras, audio sensors, vibration sensors, and light sensors will be integrated in the same sensor node. In addition, it is expected that the number of such highly capable sensor nodes in multimedia applications will scale to tenths, hundreds or even thousands [20].

Although many solutions have been proposed for the problem of congestion in WSNs [21], [22], [23], [24] during the last few years, the overwhelming majority of them inherit fundamental congestion control mechanism used in the Internet. Bearing in mind the current and future expectations in the area of WSNs, new performance control protocols are required, possessing desirable characteristics. The unpredictable nature of WSNs necessitates *robust, scalable*, and *self-adaptive* mechanisms which are vital to the mission of dependable WSNs. Due to the constrained nature of WSNs, the new approaches should be *simple* to implement at individual node level with *minimal exchange of information.* The focal point of this study is to design a *robust, scalable, self-adaptive and energy-efficient congestion control (CC) mechanisms* for delivering enhanced application fidelity at the sink in terms of packet delivery ratio and delay, under varying network conditions.

WSNs in many ways, can be likened to social groups found in nature (with nodes or packets being constituents of these social groups) attempting to accomplish their

tasks collectively (by simple neighbor-to-neighbor interactions), in a decentralized manner, and in the absence of (external) central supervision. Bio-systems usually exhibit remarkable survivability and robustness to external stimuli and internal perturbations or loss of units, as well as excellent scaling properties. Adaptation is one of bio-systems' major strengths as they must respond to addition or removal of members, as well as to sudden changes in the environment. This study explores what can be learned from the behavioral tendencies of natural systems and applies them in designing robust network control techniques. Drawing inspiration from the collective behavior of natural social groups, local behavior can be dictated easier and an emergent global behavior of minimum congestion and direction of information flow to the sink can be determined. In this way, self-* properties, e.g. self-organization and self-adaptation, are not implemented explicitly into individual devices or nodes, but emerge as a result of the design of the congestion control approach.

This study is motivated by two examples of collective behavior found in nature: (a) the obstacle avoidance and the orientation behavior of bird flocks, and (b) the population dynamics of competition and coexistence among species found in nature.

The collective intelligence of bird flocks using a few simple rules that led to efficient and effective solutions for real world problems (see next paragraph) served as the basis for moving towards a bird flocking oriented approach. Also, the inherent ability of bird flocks to manoeuvre around obstacles and the oriented movement of migratory birds towards a global attractor (i.e. the poles) were seen as prime sources of inspiration due to their similarities to the a possible way of avoiding

congestion in WSNs. In WSNs, packets are expected to move in the network towards a global attractor (the sink). In order to avoid congestion, packets should be able to manoeuvre around regions of congestions or 'dead' nodes. WSNs necessitate simple, decentralized and self-adapting solutions, and a bird flocking oriented approach has all the prerequisites for successful development of a CC approach.

Flocking behavior was first simulated on a computer in 1986 by Craig Reynolds [10] with his simulation program, Boids. As was made clear in this first attempt, the potential of the flocking behavior comes from the ability of the flock to exhibit some complex collective intelligence when just a few simple governing rules are applied to each individual in the flock (i.e. a bird). The flocking behavior was successfully employed in many research studies but also it has some exciting real world applications. More specifically, the flocking behavior was used: (a) for hypothesis development and testing [11], (b) for creating visualizations for aesthetic and artistic purposes in research [10], [25] and in movies [26], [27], (c) for solving mathematical optimization problems [28], [29], (d) in aerospace engineering (e.g. by sending UAV on missions in flocks) [30], [31], and (e) for distributed systems analysis, search, and optimization [28], [29], [32].

The Lotka-Volterra (LV) competition model is a simple and well-studied mathematical population model of biology. The model was developed independently by Alfred J. Lotka (1925) [33] and Vito Volterra (1926) [34]. This model describes

animal populations that affect each other when competing for a limited shared resource. Animals are seen as analogous to traffic flows that compete for the limited shared resources of a sensor network.

In the existing literature, the LV model was used (a) to simulate the behaviors of economic systems (e.g. predict changes in wages and employment [35]), (b) in epidemiology and immunology [36] (e.g. to model the spread of infectious diseases), and (c) in the Internet to improve the TCP congestion control mechanism [37].

1.3 Contribution

This study proposes two novel nature-inspired congestion control approaches for WSNs that are described in detail and evaluated under varying network and traffic conditions in the following chapters. This section introduces the concept of each approach.

It is worth pointing out that the two proposed approaches try to avoid and mitigate congestion on a different basis. Flock-CC guides packets to exploit alternative available paths to the sink in order to bypass the congested or 'dead' region without 'forcing' source nodes to decrease their sending rates when congestion or node failures occur. In other words, Flock-CC follows a strategy of *'fixed rate, find receiver'*. On the other hand, the LVCC approach regulates the transmission rates of source nodes through given (established) paths to the sink. Thus, LVCC follows a strategy of *'fixed receiver, find rate'*.

1.3.1 The Flock-based congestion control (Flock-CC) approach [1], [2], [3], [4], [5]

Flock-CC aims to provide congestion avoidance by mimicking the *obstacle avoidance behavior and the orientational movement of bird flocks*, where packets are modeled as birds flying over a topological space (e.g. a sensor network) whilst trying to avoid obstacles (e.g. congestion regions and areas of 'dead' nodes).

The novelty of the proposed approach lies in the use of the flocking behavior of birds, neither for aesthetic purposes, nor for solving an optimization problem, but for solving a frequent realistic problem in the area of wireless sensor networks.

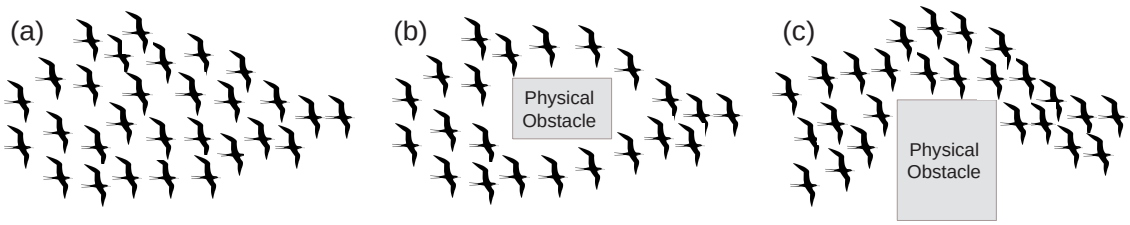


Figure 2: The obstacle avoidance behavior of bird flocks: (a) A flock of (migrating) birds moves through an obstacle-free environment (towards a magnetic pole); (b) an obstacle 'forces' the flock to split into 2 subflocks; (c) the obstacle extends further on one side and the flock is reformed along the path bypassing the obstacle on the other side. Flocking behavior was first simulated on a computer in 1986 by Craig Reynolds [10] with his simulation program, Boids.

The main idea behind this study is to move packets to the sink, whilst providing congestion control by mimicking the *obstacle avoidance behavior of bird flocks* illustrated in Fig. 2. The Flock-based Congestion Control (Flock-CC) approach is proposed, where packets are modeled as birds flying over a topological space, e.g. a sensor network. The main idea is to 'guide' packets to form flocks and flow towards

a global attractor (which is the sink in WSNs), whilst trying to avoid obstacles (failing nodes and congested regions). Inspiration for designing Flock-CC is drawn by: (a) the repulsive and attractive interactions among closely located individuals proposed by Couzin et al. [11], (b) the orientational movement of migrating birds towards a global attractor (poles or equator) under the influence of the magnetic field of Earth, and (c) the limited visual perception (field of view) of individuals within the flock.

Congestion control is carried out in a hop-by-hop manner on each node along the path to the sink using traffic spread in multiple paths. Each packet chooses its next hop node taking into account a desirability function, which synthesizes attraction and repulsion forces exercised by neighboring packets in the field of view (FoV), as well as the global attractive force to the sink.

Here's a possible real world analogy that describes the Flock-CC idea: When arriving at an airport, a passenger is expected to clear immigration. Upon entering the immigration area, the passenger observes that there are three potential queues, out of which one is completely empty, one has a few people waiting, and one has many people waiting. Which queue would you choose? One may argue that it is the second one. Why not choose the third one is clear. You would have to wait a long time. Thus, repulsion makes sense. Why not choose the first one? Because presumably there is something wrong with it, for otherwise people would have chosen that queue as well. Thus, attraction make sense.

Initial attempts for the development of the Flock-CC approach are described in [1], [2], and [3]. An improved version of the Flock-CC approach (second model) is described in [4] and [5]. This study presents and evaluates the improved (and final) Flock-CC model, which mimics more faithfully the bird flocking paradigm, as presented by Couzin at al. [11]. The final model is simpler, involving only two tunable parameters instead of four (hence easier to tune and thereafter deploy), while it maintains comparably good performance characteristics. Initial attempts on developing the Flock-CC approach are described in [38].

The Flock-CC approach differs in two aspects from Couzin's model: (1) The bio-swarm model of Couzin was formulated on the metrical (continuous three-dimensional) space, whereas the Flock-CC model is applied on a two dimensional topological (discrete) space defined by the graph of nodes. In this sense, the Flock-CC approach can be applicable in other problem domains exhibiting discrete 2D space. (2) in Couzin's model (as well as in the Reynolds' model) individuals form flocks and move constantly in a given finite space without any attraction to a global target (final destination). On the other hand, in the Flock-CC approach, packets are expected to form flocks and move towards a global attractor (sink). The latter Flock-CC characteristic necessitates the existence of a field of attraction towards the sink.

Before introducing a detailed description of the Flock-CC approach, it is worth illustrating in a simple WSN some intrinsic properties. The emergent global behavior of the Flock-CC approach can be perceived by the visual representation of flock

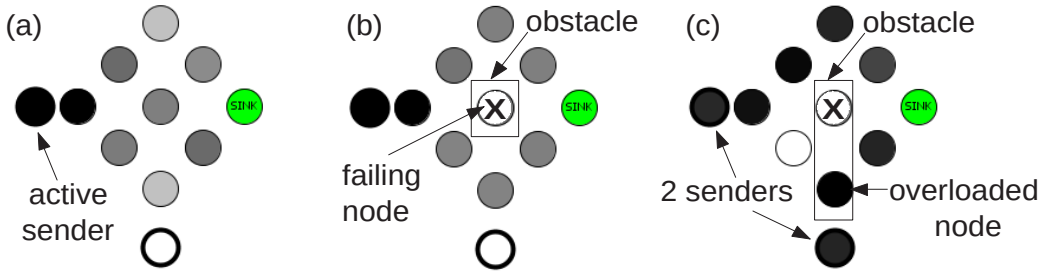


Figure 3: The obstacle avoidance behavior of packet flocks demonstrated through a minimal topology. Color intensity indicates the intensity of packets visiting within a given 1-second time slot. (a) A flock of packets moves through an obstacle-free environment towards the sink; (b) an obstacle (failing node) 'forces' the flock to split into 2 subflocks; (c) after the activation of the second sender, the obstacle is supplemented by an overloaded node which causes the packet flock coming from the back of the network to adapt and follow a less congested path to the sink (top part of the network).

movements as well as on the basis of the performance evaluation metrics. We experimented with a minimal topology (details of the simulation environment appear in Chapter 5). Fig. 3(a) shows the movement of packet flocks over a sensor network on the basis of the Flock-CC approach, mimicking the behavior of natural bird flocks of Fig. 2(a) in a non-congested and failure-free environment. In Fig. 3(b), Flock-CC exhibits the obstacle avoidance behavior of bird flocks in a failure prone environment and in 3(c) in an overloaded environment. In Chapter 5, extensive results reveal the significance of the intrinsic flocking characteristics for improving the performance of WSNs.

The main challenges for Flock-CC are to achieve: (a) low number of collisions and retransmissions, (b) low packet loss resulting in high packet delivery ratio and thus reliability and low energy tax, (c) low latency, and (d) tolerance against failures.

The Flock-CC approach primarily targets event-based WSNs used, for example, in process control and disaster recovery missions. In event-based WSNs, packet bursts can be dynamically and randomly initiated at any sensor node within the network since each node is expected to report to the sink once the occurrence of a given event has been detected. It is not uncommon for many neighboring nodes to initiate transmission at the same time in order to report an event in their vicinity.

Major contributions arising from the Flock-CC approach are summarized below. More specifically, Flock-CC:

- alleviates congestion by balancing the offered load through alternative paths to the sink,
- offers acceptable packet delivery ratio (percentage of packets delivered to the sink) especially in high load scenarios, fast delivery of packets to the sink and low energy tax,
- achieves adaptation to changing network and traffic conditions, robustness against failing nodes, scalability in different network sizes,
- outperforms typical conventional congestion-aware and congestion-free routing approaches in terms of packet delivery ratio in low, medium and high loads.

1.3.2 The Lotka-Volterra based congestion control (LVCC) approach [6], [7], [8], [9]

Population dynamics has traditionally been the dominant branch of mathematical biology which studies how species populations change in time and space and the processes causing these changes. Simple mathematical biology models [39] can be used to model the evolution in species populations. Information about population dynamics is important for policy making and planning. In this study, population dynamics are used for designing a congestion control policy.

Population dynamics can be modeled with a simple balance equation that describes how the overall population size of a species changes over time as a result of species interaction with resources, competitors, mutualists and natural enemies. The second congestion control approach focuses on the *Lotka-Volterra (LV) competition model* [33], [34], which is a deterministic competition model of mathematical biology which involves interactions among species that are able to coexist, in which the fitness of one species is influenced by the presence of other species that compete for at least one limiting resource. The LV competition model is considered to be one of the most studied mathematical models of population biology.

Based on the LV competition model, a decentralized congestion control approach is proposed that regulates the rate of every flow in order to prevent, or at least gracefully minimize congestion whilst aiming to achieve fairness among competing flows. More specifically, the *main objective is to provide efficient and smooth rate*

allocation and control while maintaining fairness and friendliness with interfering flows, and providing graceful performance degradation as the offered load increases.

The LV competition model was also applied by other researchers in modifying the congestion control mechanism of TCP by Hasegawa and Murata [37]. However, *the novelty of the LVCC approach lies in the fact that the LV model is applied to WSNs in a hop-by-hop manner.*

The LV-based congestion control (LVCC) mechanism is targeted for small-scale dependable multimedia WSNs [17] and especially for applications that require continuous stream of data.

Major contributions arising from the LVCC approach are summarized below:

- LVCC preserves the global properties of biological processes such as stability, self-adaptation, scalability and fairness, that are achieved collectively without explicitly programming them into individual nodes. Analytical evaluations and simulations were performed to understand how the variations of the model's parameters influence stability, sensitivity to parameters, scalability and fairness.

- Control system type simulations in Matlab validated the correctness of analytical results for plausible scenarios that could not be formally tested.

- Simulations showed that the proposed model achieves stability and smooth network operation under the analytically proposed conditions. Realistic scenarios of network operation and conditions were also simulated for effective parameter setting.

- Realistic scenarios evaluation suggested certain values for parameters α, β and r that are able to achieve high packet delivery ratio, low end-to-end delay, scalability and fairness among competing flows.

- LVCC was found to outperform AIMD-like rate-based congestion control approaches for WSNs in terms of stability and flow rate smoothness.

1.4 Publications list

This section provides a list of all publications stemming out of this thesis. Major contributions of this thesis have been published in:

Journal papers

1. Pavlos Antoniou and Andreas Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka-Volterra competition model," Elsevier Computer Communications, Special Issue on Applied Sciences in Communication Technologies, vol. 33, no. 17, pp. 2039 – 2047, November 2010.

2. Charalampos Sergiou, Pavlos Antoniou, and Vasos Vassiliou, "Congestion control protocols in wireless sensor networks: A survey," submitted to the IEEE Communications Surveys and Tutorials (accepted, subject to minor revision), 2012.

Book chapters

1. Pavlos Antoniou and Andreas Pitsillides, "Congestion control in wireless sensor networks based on the Lotka Volterra competition model," in Biologically Inspired Networking and Sensing: Algorithms and Architectures, D. C. Verma and P. Lio, Eds. IGI Book, August 2010, pp. 158 – 181.

Conference and Workshop papers

1. Pavlos Antoniou, Andreas Pitsillides, Andries Engelbrecht and Tim Blackwell, "Applying swarm intelligence to a novel congestion control approach for wireless sensor networks," Invited paper, 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2011, October 2011.

2. Pavlos Antoniou, Andreas Pitsillides, Andries Engelbrecht and Tim Blackwell, "Mimicking the bird flocking behavior for controlling congestion in sensor networks," Invited paper, 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2010, November 2010.

3. Pavlos Antoniou, Andreas Pitsillides, Andries Engelbrecht, Tim Blackwell, and Loizos Michael, "Congestion control in wireless sensor networks based on the bird flocking behavior," in 4th IFIP TC 6 International Workshop on Self-Organizing Systems IWSOS, ser. Lecture Notes in Computer Science, T. Spyropoulos and K. A. Hummel, Eds., vol. 5918. Springer, December 2009, pp. 220 – 225.

4. Pavlos Antoniou and Andreas Pitsillides, "Congestion control in autonomous decentralized networks based on the lotka-volterra competition model," in 19th International Conference on Artificial Neural Networks ICANN, Part II, ser. Lecture Notes in Computer Science, C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinas, Eds., vol. 5769. Limassol, Cyprus: Springer, September 2009, pp. 986 – 996.

5. Pavlos Antoniou, Andreas Pitsillides, Tim Blackwell, and Andries Engelbrecht, "Employing the flocking behavior of birds for controlling congestion in autonomous decentralized networks," in 2009 IEEE Congress on Evolutionary Computation, A. Tyrrell, Ed., IEEE Computational Intelligence Society. Trondheim, Norway: IEEE Press, May 2009.

6. Pavlos Antoniou and Andreas Pitsillides, "Towards a scalable and self-adaptable congestion control approach for autonomous decentralized networks," in 3rd European Symposium on Nature-inspired Smart Information Systems (NiSIS 2007), St. Julians, Malta, November 2007.

Technical Reports

1. Pavlos Antoniou, Andreas Pitsillides, Tim Blackwell, Andries Engelbrecht, and Loizos Michael, "From bird flocks to wireless sensor networks: A congestion control approach," Department of Computer Science, University of Cyprus, Tech. Rep. TR-11-5, September 2011. [Online]. Available: http://www.cs.ucy.ac.cy/csp5pa1/publications/TR-11-5.pdf

2. Pavlos Antoniou and Andreas Pitsillides, "Understanding complex systems: A communication networks perspective," Department of Computer Science, University of Cyprus, Tech. Rep. TR-07-01, February 2007. [Online]. Available: http://www.cs.ucy.ac.cy/csp5pa1/publications/TR-07-01.pdf

Submitted

1. Pavlos Antoniou, Andreas Pitsillides, Tim Blackwell, Andries Engelbrecht, and Loizos Michael, "Congestion control in wireless sensor networks based on bird flocking behavior," submitted to the Elsevier Computer Networks Journal, 2012.

The research on the flocking behavior of birds was guided by Dr. Andreas Pitsillides (Professor, Department of Computer Science, University of Cyprus), with contributions from Dr. Andries Engelbrecht (Professor, Department of Computer Science, University of Pretoria, South Africa), Dr. Tim Blackwell (Senior Lecturer, Department of Computing, Goldsmiths College, University of London, UK) and Dr. Loizos Michael (Lecturer, Open University of Cyprus). The research on the competitive coexistence behavior of species was guided by Dr. Andreas Pitsillides. In addition, a survey on congestion control was carried along with Mr. Charalampos Sergiou (Ph.D.student, Department of Computer Science, University of Cyprus) and Dr. Vasos Vassiliou (Assistant Professor, Department of Computer Science, University of Cyprus).

1.5 Structure of the report

This thesis is organized as follows: Chapter 2 gives background information on the problem of congestion in WSNs, introduces both the novel paradigm of Swarm Intelligence (with emphasis on the flocking behavior of birds) and the Lotka-Volterra competition model as well as presents previous work on congestion control for WSNs. Chapter 3 addresses congestion in event-based applications, and presents the design of the proposed flock-based congestion control approach. Chapter 4 focuses on congestion in streaming applications and presents the Lotka-Volterra based congestion control approach. In Chapter 5, performance evaluation results are presented. Finally, Chapter 6 concludes this study and proposes areas of further work.

Chapter 2

Background

2.1 Congestion in WSNs

In this section, the problem of congestion is discussed with respect to the different types of WSNs applications. Next, the role of medium access protocols in congestion is discussed on the basis of two prominent types of medium access control protocols. Furthermore, this section presents the consequences of congestion supplemented by the different types of congestion as well as where congestion occurs. The last two subsections deal with congestion detection indicators that are mainly used in WSNs as well as with congestion avoidance and mitigation techniques.

2.1.1 The problem

A WSN consists of spatially distributed autonomous sensor nodes that are able to monitor physical or environmental conditions, such as temperature, humidity, sound, vibration, pressure, motion or pollutants and to cooperatively pass their

data through the network to one or more sink nodes. Sink nodes are responsible for gathering data and forwarding them to the outer world, e.g. the Internet, for further processing.

WSNs are currently being employed in a plethora of applications ranging from medical to military, and from home to industry. All WSN applications can be categorized under three data delivery models: (a) event-based, (b) continuous-based (streaming) and (c) query-based. In this thesis, we focus on event-based and continuous-based (streaming) data delivery models.

Event-driven applications: Typically, event-driven WSNs operate under light load but large, sudden, and correlated-synchronized impulses of data may suddenly arise in response to a detected or monitored event. Most event-driven applications (e.g. target tracking, fire detection) are interactive, delay intolerant (real-time) and mission critical. That means that data generated from sensor nodes should be delivered within short span of time through a sink node to a processing center for further actions.

The data traffic generated by a single sensor node may be of very low intensity. However, very bursty traffic may be generated by a set of sensors due to a common event or a phenomenon. In addition, the converging (many-to-one) nature of packets from multiple sending nodes to one or more sink nodes may lead to congestion around the sinks.

Continuous-driven (streaming) applications: Over the last few years, WSNs are being developed towards a large number of multimedia streaming applications,

e.g. video surveillance, traffic control systems, health monitoring, and industrial process control. In streaming applications, sensor nodes send their data continuously to the sink at a constant or regulated (by a formula) rate. Streaming applications may server real-time or non-real-time data. Real-time data is delay-constrained and has a certain bandwidth requirement. Packet losses can be tolerated to a certain extent. On the other hand, non-real-time data can be send when the sink may want to collect periodic data from the sensor field. In this context, delay and packet losses are both tolerated.

In streaming applications, the increasing reporting rate of nodes, perhaps due to demand of higher data fidelity, in conjunction with the uncontrolled use of scarce network resources may lead to congestion. More specifically, packet flows left uncontrolled (i.e. to reach high sending rates, or to use the same paths to the sink) are likely to cause congestion even if local contention is minimized.

2.1.2 The role of Medium access control (MAC) protocols in congestion

When two neighboring sensor nodes attempt to access the wireless communication channel (shared medium) simultaneously, contention takes place leading to collision. Collision is a symptom of congestion in the wireless channel and can result in a time-variant channel capacity. Therefore, a medium access control (MAC) protocol is needed to to co-ordinate the access of nodes to the shared medium so that packets transmitted from the sensor nodes will not interfere with each other.

Various MAC protocols with different objectives were proposed for wireless sensor networks. Two major types of media access control protocols are prevalent in WSNs: Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

TDMA is a schedule-based, contention-free protocol. TDMA divides the channel into time slots and only one node is allowed to transmit or receive data within each time slot. In this way, TDMA does not allow any collisions and bounds the delay. However, the major advantage of TDMA is its energy efficiency, because it directly supports low-duty-cycle operations on node. On the other hand, TDMA assumes that the sensor nodes are time-synchronized, which is not possible for large scale sensor networks due to the significant amount of signaling traffic needed. Also, TDMA protocols have limited scalability and adaptivity to the changes on number of transmitting nodes. When nodes become active or nodes stop transmitting, the slots should be reallocated according to the number of transmitting nodes. Frequent changes may be expensive or slow to take effect. Also, static slot allocation can limit the available throughput for any given node, and the the maximum number of active nodes may be limited.

CSMA/CA is a contention-based, random access MAC protocol, that requires no coordination among the nodes accessing the channel. In CSMA/CA, when a sensor nodes has a packet to transmit, it senses the channel before transmitting and is allowed to transmit after the channel is sensed idle. If a packet is successfully received, the destination node issues an acknowledgement packet. The asynchronous

mechanism of accessing the shared medium is not able to prevent collisions. When a collision is detected, the lost packet is retransmitted by the MAC layer after a random amount of time (back-off procedure). If the problem persists, the colliding packet is discarded after a finite number of unsuccessful retransmission attempts. CSMA/CA protocol employs a retransmission scheme in order to increase the reliability of the lossy wireless channel. The main advantage of CSMA/CA is that it adapts quite well with the variable condition of traffic and is quite robust against interferences. The main drawback of CSMA/CA is that it consumes more energy than contention-free protocols because of energy wastage in collisions and idle listening. Moreover, CSMA/CA does not provide bandwidth and delay guarantees. The well-known WiFi (i.e. IEEE 802.11) protocol follows is a CSMA/CA-like protocol.

The majority of MAC protocols designed for WSNs rely on the CSMA/CA protocol [40] and thus, **both approaches proposed in this thesis assume a CSMA/CA-based underlying MAC protocol** with a maximum number of 7 retransmissions before discarding a recurring colliding packet. CSMA/CA has been preferred to TDMA as medium access scheme for WSNs in the scenarios where the traffic is bursty and multiple consecutive and contiguous packets generated from the same collision neighborhood need to be sent. TDMA can be used in scenarios where a small set of nodes is continuously transmitting packets with infrequent activation/deactivation of transmitting nodes.

2.1.3 Consequences of congestion

Congestion may cause increase in collisions and retransmissions at the CSMA/CA-based MAC layer (multiple packet losses), low link utilization (throughput reduction), and increase of queueing delays, leading to the deterioration of the offered quality of service (QoS). Increasingly, congestion in WSNs is responsible for energy waste, decrease of network lifetime and even for the decomposition of network topology in multiple components.

2.1.4 Types of congestion

In traditional Internet wired networks, packet loss is taken as an indication of congestion while congestion control is usually carried out in an end-to-end manner (i.e. only the source-destination pair is involved). In these networks, packet loss is solely attributed to buffer overflows as a result of finite buffer queues.

Wireless communication has significantly different characteristics compared to wired networks such as higher bit error rates, higher latency, limited bandwidth, multi-path fading of the signals and handoff. The problem is worsened in multi-hop wireless networks, where communication between two end nodes is carried out through a number of intermediate nodes whose function is to relay information from one point to another. In wireless multi-hop networks like WSNs, packet losses can be mainly attributed to either *buffer overflows*, or *collisions in the wireless medium* when more than one nodes are trying to access the channel simultaneously. A smaller but significant amount of packet losses can also occur randomly due to

environmental factors as for example, interference from other devices, node mobility, bad channel conditions, as for example multipath fading, disconnections due to coverage limitations etc.

Based on the two aforementioned congestion symptoms in WSNs, congestion phenomena can be categorized into two types:

1. **Buffer-type**: When the incoming traffic load exceeds the outgoing channel capacity at a particular node, packets are being accumulated in the node's buffer. If the problem persists, instantaneous queue length exceeds the buffer capacity leading to buffer overflow, and long delays.

2. **Link-type**: The multi-hop nature of WSNs, the shared communication medium and the limited bandwidth give rise to link-type congestion. In wireless networks, local channel contention arises in the vicinity of a sensor node due to the limited bandwidth and interference among multiple neighboring sensor nodes that try to access the wireless medium simultaneously. The wireless channel contention and interference that occur between different flows or between packets of a flow can result in a time-variant channel capacity in the area of node. This time-variant nature makes the congestion level fluctuating and unpredictable even in case of constant incoming traffic rate. The problem is worsened when densely deployed sensor network topologies are considered that exacerbate the impact of the channel contention. Even though the high number of event sources may improve event detection efficiency, closely located source nodes are expected to increase wireless channel contention.

2.1.5 Where congestion occurs?

In densely deployed sensor networks, a large number of data packets are generated from nodes located in the area of the event that report the same event concurrently. In this way, hotspots are created very close to source nodes leading to the exhaustion of the locally available resources (wireless channel capacity, buffer capacity of involved nodes).

Even in sparsely deployed sensor networks, when source nodes report at high data rates, hotspots may arise around a sink node due to the converging (many-to-one) nature of packets from multiple sending nodes to a single sink node. Furthermore, in sparsely deployed networks, even when source nodes report at low data rates, hotspots may occur anywhere but likely further from the sources, towards the sink as a result of two or more intersecting traffic flows. Also, a clusterhead in a cluster-based topology or a parent node with many children in a tree-based topology are potential hotspot points.

From the earlier discussion it is clear that congestion can occur due to random causes and also appear in many parts of the network.

2.1.6 Congestion detection

One of the main problems in designing a congestion control strategy for WSNs is how to detect congestion. In the Internet, conventional congestion detection techniques depend heavily on packet loss due to buffer overflows to infer congestion, and to a lesser extend on queue occupancy, and end-to-end delay.

Simulation studies conducted by [21] and [22] revealed that, in WSNs where the wireless medium is shared using CSMA/CA-like protocols, wireless channel contention losses can dominate buffer drops and increase quickly with offered load. The problem of channel losses is worsened around hot spot areas, as for example, in the proximity of an event, or around the sink (see next subsection). These phenomena result in the starvation of channel capacity in the vicinity of senders, while the wireless medium capacity can reach its upper limit faster than queue occupancy [41]. Thus, *queue occupancy alone cannot accurately serve as an indication of congestion.* On the other hand, the study of Hull et al. [22] in a large sensor network testbed revealed that when used alone, wireless channel-based congestion detection performs worse than queue occupancy-based congestion detection. Therefore, congestion is expected to be more effectively detected using a combination of buffer occupancy and wireless channel load.

However, in small scale networks, or in networks with predefined deployment, schedule-based medium access control protocols such as TDMA may be able to resolve the problem of collisions. Therefore, the problem of buffer overflows is considered to be more critical in WSNs than the classical Internet due to buffer size limitations. Even at low traffic rates, buffer overflows can be experienced at some point of the network (usually close to the sink) due to the converging (many-to-one) nature of packets from multiple sending nodes to a single sink node. The problem of buffer overflows can be somehow minimized user larger buffer spaces (if the constrained memory of sensor nodes is sufficient) at the cost of larger delays.

Currently, three congestion detection indicators are mainly used in WSNs, which are presented below:

2.1.6.1 Buffer occupancy

Some congestion control approaches for WSNs as for example, [23], [42], [43], and [44] are solely based on buffer occupancy to detect congestion. In these approaches, congestion is detected when the instantaneous queue length of a node exceeds its limited buffer capacity, leading to packet drops, or when the queue length exceeds a certain percentage of the buffer capacity, leading to long delays. These approaches assume that the underlying MAC protocol can efficiently provide a stable radio link capable of resolving collisions. For example, some approaches like [42] assume perfect MAC.

2.1.6.2 Buffer occupancy and wireless channel load

A large number of congestion control approaches in WSNs, as for example CODA [21], Fusion [22], BGR [45] and Siphon [46] use both buffer occupancy and local wireless channel load to infer congestion. In these approaches, while a packet waits to be sent, the sensor node samples the state of the channel at a fixed interval. Based on the number of times the channel is busy, it calculates a utilization factor. If utilization rises above a certain level (e.g. the theoretical upper bound of the channel throughput), the congestion bit is set. Otherwise, the congestion bit is cleared.

2.1.6.3 Packet inter-arrival and service time

Recent studies by [47], [48], [49] and [50] argue that congestion can be also inferred by inspecting packet inter-arrival time and packet service time (or alternatively incoming and outgoing traffic rates). In these approaches, congestion is inferred when the inter-arrival time is smaller than the service time, that is the incoming packet rate is higher than the outgoing traffic rate leading to accumulation of packet in queues.

In this thesis, buffer occupancy and wireless channel load are cooperatively used to infer congestion.

2.1.7 Congestion notification

After congestion detection, somehow the rest of the network should be notified in order for nodes to take measures for congestion mitigation. Congestion notification can be divided into two categories as shown below:

2.1.7.1 Explicit

In explicit congestion notification, nodes which detect congestion broadcast control packets in order to inform some (or all) nodes in the network about the event. Early studies for congestion control in WSNs [21], [23], used this explicit way of informing about congestion. However, it can be argued that this method increases significantly the amount of traffic injected into the network, exacerbating the problem of congestion.

2.1.7.2 Implicit

Implicit congestion notification dominates in the latest congestion control approaches in WSNs. In implicit notification, no additional control messages are needed to propagate congestion information. In a number of CC approaches [23], [21], [22], congestion information is piggybacked on normal data packets. This information can be a single bit on the data packet header [23]. In other approaches like the proposed approach, congestion is implicitly inferred by each node located inside a congestion region without any other node, involved in the process.

2.1.8 Congestion control: avoidance and mitigation

Congestion control (CC) is concerned with measures taken in order for a network (a) to avoid congestion (congestion avoidance approaches) and (b) to mitigate congestion (congestion mitigation approaches) and operate within an acceptable performance level, even when demand is near or exceeds the capacity of network resources. Congestion control is a global issue which involves every node within a network. Congestion control is concerned with efficiently using a network at any load.

Congestion control approaches in WSNs should monitor and control traffic flows and network resources so as to avoid congestive collapse by attempting to detect and avoid oversubscription of any of the buffer or wireless channel capabilities of sensor nodes and taking appropriate steps to alleviate the problem. Congestion control can be primarily achieved through traffic manipulation (e.g. rate adaptation to network

changes, routing of packets through multiple paths, topology control in terms of clustering formation), and network resource management (e.g. power control, multiple radio interfaces). It is beyond any doubt that without congestion control, a network can easily become gridlocked, with little or no data being transported from source nodes to the sink.

End-to-end CC approaches used in the Internet are not expected to be effective in multi-hop, error prone wireless environments because the end-to-end nature may result in reduced responsiveness causing increased latency and high error rates, especially during long periods of congestion. Due to their severely constrained nature, WSNs necessitate autonomous, decentralized, fast time scale congestion control strategies which promise immediate, effective and efficient congestion avoidance or relief from congestion. Decentralized approaches are expected to adopt a hop-by-hop model where all nodes along a network path can be involved in the procedure. Each node should make decisions based only on locally available information (e.g. buffer load, channel load) since it is not practical for the nodes to have complete information about the system state. This is a highly desirable feature as it minimizes the exchange of messages, hence improves both energy and congestion.

2.2 Swarm intelligence and the flocking behavior of birds

2.2.1 Biological (natural) swarms: The bird flock paradigm

This study draws inspiration from nature, which has been very successful in effectively solving similar types of complex problems. Recently, nature-inspired computing has been fueled by the emergence of a novel computational paradigm, the so-called Swarm Intelligence (SI) paradigm [28], [32]. A swarm is an apparently disorganized collection (population) of moving individuals that tend to cluster together while each individual seems to be moving in a random direction[1] . SI techniques, motivated by the collective behavior of biological (or natural) swarms (e.g. bird flocks, ant colonies, fish schools) living in decentralized, self-organizing, and adapting environments, reportedly provide a promising basis for computing environments that need to exhibit these characteristics [12], [51], [52] and [53]. Research in SI has provided computer scientists with powerful methods for designing distributed control and optimization algorithms. These methods are applied successfully to a variety of scientific and engineering problems [54]. In addition to achieving good performance on a wide spectrum of 'static' problems, swarm-based algorithms tend to exhibit a high degree of flexibility and robustness in dynamic environments [54].

Biological swarms found in nature carry out their tasks collectively in order to accomplish a given task in a complex world (e.g. foraging, migration, nest building, defence against predators). More specifically, they organize themselves into remarkable, beautiful spatiotemporal structures in a process known as self-organization.

[1]This description was adapted from a presentation by R. C. Eberhard

This organization is thought to arise from the instantaneous dynamics of individuals within a swarm, and not by any central leadership. In addition, an individual has only local perception of the surrounding environment and exhibits specific behavioral tendencies which are governed by a few simple rules. Individuals communicate with each other through the modification of the environment in a biological process known as stigmergy.

The collective global behavior of a swarm emerges from simple interactions among the constituent entities. These interactions can be viewed at different levels: the microscopic level describes the entities involved and their behavior on the basis of simple rules; the macroscopic level describes the (emergent) behavior of the overall system. The global behavior of a swarm is emergent because the rules followed by the constituents of the swarm do not contain any notion of the whole.

Mark Millonas [55], has articulated five basic principles of Swarm Intelligence:

1. Proximity principle: the population should be able to carry out simple space and time computations

2. Quality principle: the population should be able to respond to quality factors in the environment

3. Diverse response principle: the population should not commit its activities along excessively narrow channels

4. Stability principle: the population should not change its mode of behavior every time the environment changes

5. Adaptability principle: the population must be able to change behavior mode when it"s worth the computational price

Note that stability and adaptability can be seen as opposite sides of the same coin.

Figure 4: A bird flock with the characteristic 'V' shape formation.

Bird flocking is a particularly familiar natural example of Swarm Intelligence. The aggregate motion of a flock of birds is a beautiful and familiar part of the natural world. A bird participating in a flock only adjusts its movements to coordinate with the movements of its flock mates, typically its neighbors that are close to it in the flock [56]. Figure 4 shows a typical bird flock formation. It can be seen that the amazing thing about the flock is that for long flight it's usually organized in a 'V' shape, which can be proven to be the most energy-efficient formation. The bird in front generates air movements that lessens the load of the birds behind. When the front bird gets tired, some bird from behind takes over.

Natural flocks seem to consist of two balanced, opposing behaviors: a desire to stay close to the flock and a desire to avoid collisions within the flock [10]. Each bird does not take commands from any leader bird since there is no leading bird. Any

bird can be in the front, center and back of the swarm. The basic urge to join a flock seems to be the result of evolutionary pressure from several factors: protection from predators, statistically improving survival of the (shared) gene pool from attacks from predators, advantages for social and mating activities, and profiting from a larger effective search pattern in the quest for food (essentially each bird is exploiting the eyes of every other bird) [57].

There is no evidence that the complexity of natural flocks is bounded in any way. When a new bird joins, the flocks do not become 'full' or 'overloaded'. It was observed that natural flocks seem to behave in exactly the same fashion over a huge range of flock populations. In addition, it does not seem that an individual bird can be paying much attention to each and every one of its flockmates. But in a huge flock spread over vast distances, an individual bird must have a localized and filtered perception of the rest of the flock. A bird might be aware of three categories: itself, its two or three nearest neighbors, and the rest of the flock [58].

Another important feature of natural bird flocks is that when the flock meets an obstacle, such as a tall building, it finds a perfect way to split apart to go around an obstacle and reform again once the obstacle is passed. The proposed flock-based congestion control approach presented in Chapter 3 exploits this behavior and features the properties discussed above.

2.2.2 Artificial swarms

Artificial swarms are the software equivalent of these remarkable biological swarms (e.g. flocks of birds) found in nature. An artificial swarm consists of a number of artificial entities, often called particles (i.e. the individuals within a social group like a bird in a flock, an ant in a colony, a buffalo in a herd). At a more abstract level, an artificial swarm exists as a set of local rules, or interactions between artificial entities. The set of rules and interactions follow the theoretical models of biological swarms. An artificial swarm may be visualized, keeping in mind that the emergent pattern formations depend on local interactions among artificial entities.

The centralized approach to simulate artificial swarms is to formulate the collective behavior as a script which each artificial entity must obey. In this way, swarming behavior is not emergent because it is built into the script from the outset. However, the famous study of Craig Reynolds[2] [10] supported the hypothesis that swarms and flocks are self-organizing when local, decentralized rules are followed by each artificial entity.

[2]Craig Reynolds realized that the coordinated motion of a flock of birds could be modeled by applying three simple rules to be followed by simulated flocking creatures, the so-called 'boids'. More details on this approach are found in Section 2.2.4.

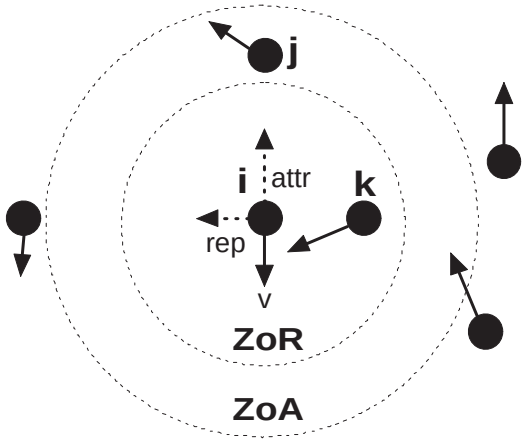


Figure 5: Rules governing the interactions among neighboring particles in a swarm: Particle i currently at x and moving with velocity v, is attracted to particle j located inside Zone of Attraction (ZoA) and repelled from particle k located inside Zone of Repulsion (ZoR). The rest of the particles are outside i's perception.

The study of artificial swarms involves reference to spatial neighborhoods around individual particles. In particular, it is assumed that individuals have a finite range of perception in which a given individual feels the influence of neighbors. Typically, as shown in Figure 5, *individuals repel each other at close range, attract each other at medium range and are oblivious to each other at long range.* The attractions provide coherence, maintaining a shared neighborhood (which may be a sub-swarm or the entire swarm) and the repulsions prevent collisions between nearby particles. The attractive and repulsive accelerations are the analogues of positive and negative feedback. At its simplest, a swarm algorithm considers the individual swarming participants as purely dynamic entities. These entities are represented as point

particles in d-dimensional real space with dynamic state (x, v). The basic rules governing the interactions between neighboring particles in a swarm or flock are:

1. if apart, move closer (cohesion)

2. if too close, move apart (separation)

3. attempt to match velocities (alignment)

The final rule only applies to collectives where their entities move in unison, such as flocks, herds and schools. Swarming entities have more chaotic motions and drop the rule of alignment.

2.2.3 Classification of contemporary swarm algorithms

A classification of contemporary swarm algorithms was raised by Blackwell [25]. In accordance with [25], contemporary swarm algorithms can be split into three groups (in order of faithfulness to natural swarms), although there are overlaps, as follows:

1. Bio-swarms are used to develop scientific models of natural systems (for example the refined bio-swarm of Couzin et al. [11] and the prey-flock escaping model of Lee et al. [59]). These swarms may be visualized, but are primarily intended for hypothesis development and testing.

2. Simulation swarms are visualizations for aesthetic and artistic purposes and do not need to accurately represent nature (Reynolds [10], Blackwell [25]). These swarms move in real time so that the visualizations have a sense of realism.

3. Social swarms use an information network rather than a spatial region to define a neighborhood for interactions. Social swarms are frequently used to solve mathematical problems, as in ant colony optimization (ACO), Bonabeau et al. [28], and particle swarm optimization (PSO), Kennedy et al. [60]. These swarms have the loosest connection to nature: the visualizations take secondary importance to the algorithmic details and in fact they can look quite unrealistic.

In the rest of this study, this terminology is followed.

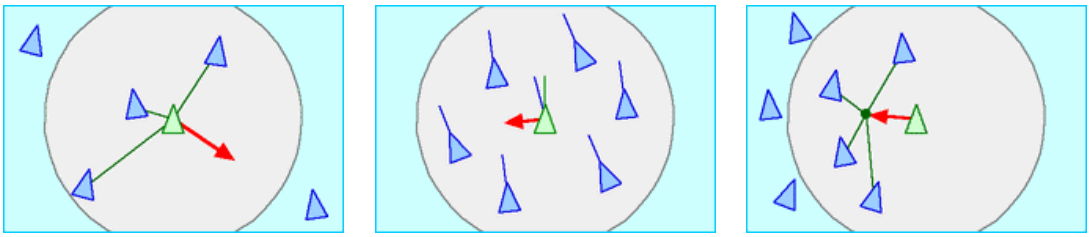
2.2.4 The 'Boids' model of Reynolds [10]

Boids is an artificial life program, developed by Craig Reynolds in 1986, which simulates the flocking behavior of birds. The idea was published in a seminal paper that appeared in the ACM SIGGRAPH conference in 1987 [10].

The model involves behaviors that correspond to the opposing forces of collision avoidance and the urge to join the flock. More specifically, the bird flocking 'boid' model consists of three simple steering behaviors which describe how an individual bird maneuvers based on the positions and velocities of its nearby flock-mates [10]:

1. Collision Avoidance (Separation): steer to avoid crowding nearby flock-mates so as to prevent collisions.

2. Velocity Matching (Alignment): attempt to match velocity with nearby flock-mates (steer towards the average heading of nearby flock-mates).



(a) Separation (b) Alignment (c) Cohesion

Figure 6: The three steering behaviors in the Boids' model of Reynolds [10].

3. Flock Centering (Cohesion): attempt to stay close to nearby flock-mates (steer to move toward the average position of nearby flock-mates).

These three steering behaviors are illustrated in Fig. 6.

Each bird has direct access to the whole scene's geometric description, but flocking requires that it reacts only to flock-mates within a certain small neighborhood around itself. The neighborhood is characterized by a distance (measured from the center of the bird) and an angle, measured from the bird's direction of flight. Flock-mates outside this local neighborhood are ignored. The neighborhood could be considered of limited perception. This process happens millisecond by millisecond. As birds fly together, individuals within the flock make decisions resulting in the collective direction the flock will travel. A flock of birds moves like a well-choreographed dance troupe. They can veer to the left in unison, and then suddenly they may all dart to the right and swoop down toward the ground, coordinating their actions very well.

The boids framework is often used in computer graphics, providing realistic-looking representations of flocks of birds and other creatures, such as schools of fish or herds of animals.

2.2.5 The Bio-swarm model of Couzin et al. [11]

Couzin et al. [11] proposed a self-organizing model of group formation in three-dimensional space which is intended for investigating the spatial dynamics of animal groups such as fish schools and bird flocks. The proposed model simulates the behavior of individuals in a bio-swarm as resulting from local repulsion, alignment and attractive tendencies based upon the position and orientation of individuals relative to one another.

The behavioral rules followed by individuals are:

1. Individuals attempt to maintain a minimum distance between themselves and others at all times. This rule has the highest priority and corresponds to a frequently observed behavior of animals in nature (Krause and Ruxton [61]).

2. If individuals are not performing an avoidance manoeuvre (rule 1) they tend to be attracted towards other individuals (to avoid being isolated) and to align themselves with neighbors (Partridge and Pitcher [62], Partridge [63]).

These behavioral tendencies are simulated using local perception and simple response behaviors.

As shown in Fig. 7, bio-swarms use three concentric zones; the rule of cohesion applies in the outer zone, alignment applies in a middle zone and at short distances

the rule of separation dominates. Individuals in bio-swarms may also have a 'blind volume' in which neighbors are undetectable.

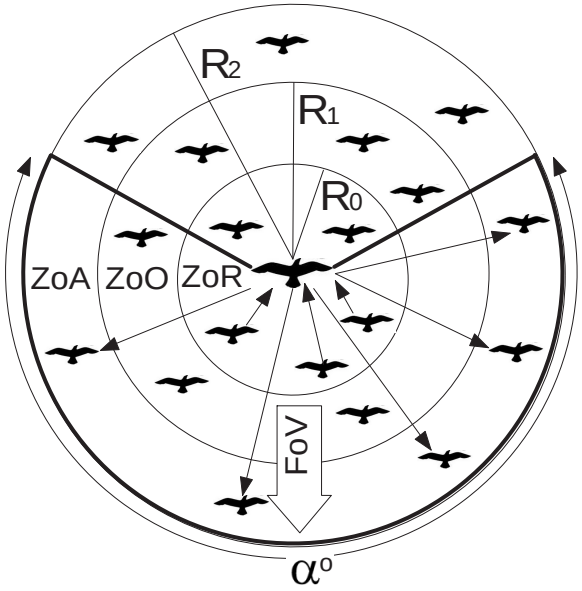


Figure 7: Representation of an individual in the model centered at the origin: ZoR =zone of repulsion, ZoO =zone of orientation, ZoA =zone of attraction. The possible "blind volume" behind an individual is also shown. α =field of perception.

In order to define the attraction and repulsion forces among individuals, some basic notations are introduced. Each individual i has a position vector c_i, and unit direction vector v_i. Time is partitioned into discrete time steps t with a regular spacing τ. In each time step, individuals assess the position and/or orientation of their neighbors within three nonoverlapping behavioral zones described above. This information is used to determine a desired direction for each individual for the successive time step $d_i(t + \tau)$ using the following rules.

The 'zone of repulsion' (ZoR) is modeled as a sphere (with radius r_r) within which each individual attempts to maintain a minimum distance from others. If n_r neighbors are present in the ZoR at time t, individual i responds by moving away

from neighbors within this zone:

$$d_r(t+\tau) = -\sum_{j \neq i}^{n_r} \frac{r_{ij(t)}}{|r_{ij}(t)|}, \tag{1}$$

where $r_{ij} = (c_j - c_i)/|c_j - c_i|$ is the unit vector in the direction of neighbor j.

If no neighbors are within the zone of repulsion, the individual responds to others within the 'zone of orientation' (ZoO) and the 'zone of attraction' (ZoA). These zones are spherical, except for a volume behind the individual within which neighbors are undetectable. This 'blind volume' is defined as a cone with interior angle $(360 - \alpha)^o$, where α is defined as the field of view (FoV). An individual with $\alpha = 360^o$ can respond to others in any direction within the behavioral zones. An individual will attempt to align itself with neighbors within the zone of orientation, giving:

$$d_o(t+\tau) = \sum_{j=1}^{n_o} \frac{V_{j(t)}}{|V_j(t)|}, \tag{2}$$

and towards the positions of individuals within the zone of attraction:

$$d_\alpha(t+\tau) = \sum_{j \neq i}^{n_\alpha} \frac{r_{ij(t)}}{|r_{ij}(t)|}. \tag{3}$$

Birds cannot see too far, thus there is no interaction with neighbors located outside the ZoA.

The attraction represents the tendency of organisms to join groups and to avoid being on the periphery, whereas the orientation allows collective movement by minimizing the number of collisions between individuals.

An analytical analysis of the model on the basis of changing parameters values revealed the existence of major group-level behavioral transitions related to minor

changes in individual-level interactions. More particularly, the bio-swarms model exhibits several collective behaviors (swarm, torus, dynamic parallel group, highly parallel group) with sharp transitions between them. Biologically, the transitions are important in allowing animal groups to change from one type of group structure to another in response to internal (e.g. hunger) or external (e.g. detection of a predator) stimuli, as individuals attempt to maximize their fitness as circumstances change.

2.2.6 Basic characteristics of the bird flocking behavior: The building blocks of the Flock-CC

The Flock-CC approach involves reference to artificial bird flocks consisting of individuals with the same behavioral model (homogeneity) with finite range of view (perception). These individuals interact with each other as well as with the environment. The design of the interactions in packet groups is influenced by the bio-swarm model of Couzin et al. [11]. The behavior of each individual is influenced by other individuals within its neighborhood.

Three inherent characteristics of bird flocks are presented next. These characteristics served as the basis for developing the Flock-CC approach.

2.2.6.1 Repulsion and attraction zones and forces in bird flocks

The notions of repulsion and attraction zones in bird flocks, as modeled by the study of Couzin et al. [11], can be used in the Flock-CC approach in order to simulate the local interactions among individuals in the flock.

However, the Flock-CC approach differs in two aspects from Couzin's model: (1) The bio-swarm model of Couzin was formulated on the metrical (continuous three-dimensional) space, whereas the Flock-CC model is applied on a two dimensional topological (discrete) space defined by the graph of nodes, and (2) in Couzin's model (as well as in the Reynolds' model) individuals form flocks and move constantly in a given finite space without any attraction to a global target (final destination). On the other hand, in the Flock-CC approach, packets are expected to form flocks and move towards the sink. The latter Flock-CC characteristic necessitates the existence of a field of attraction towards the sink. These characteristics are discussed in the next subsection.

Motivated by Couzin's model, the collective behavior of the flock is considered an emergent behavior arising from a few *simple behavioral rules* that are followed by individuals, such as:

1. repel from neighbors (if too close, i.e. within the ZoR) to avoid collisions,
2. attract to neighbors (if apart, i.e. within the ZoA) to maintain coherence among the members of a flock,
3. match velocity (speed and direction) with neighbors (within the ZoO),
4. introduce a randomness that allows for exploration of alternative paths.

These behavioral rules govern individual-level interactions which collectively result in the emergence of group-level transitions.

2.2.6.2 Birds vision and the field of view

The motion of each individual in the flock is only influenced by its nearest flock mates. Therefore, vision is the most important sense for birds. *The field of view (FoV) determines the extent of the observable world that is seen by each bird at any given moment.* Depending on the placement of the eyes, different animals have different fields of view. Some birds have a 360^o FoV.

As shown in Fig. 8, the range of visual abilities is not uniform across a field of view, and varies from animal to animal [3] .

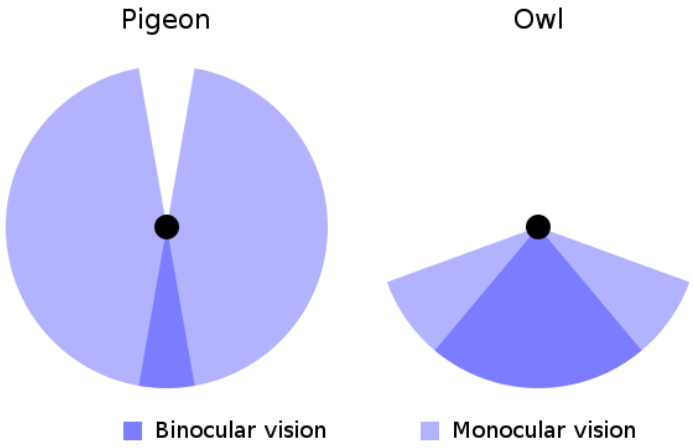


Figure 8: Fields of view for an owl and a pigeon.

2.2.6.3 Magnetic fields and birds orientation

The aforementioned behavioral rules and the presence of the field of view do not imply that a bird flock will establish a flight path towards a specific attractor. The

[3]The type of vision in which each eye is used separately is called monocular vision. The vision that occurs when the field of vision from each eye overlaps is called binocular vision, and it is quite important for depth perception. Monocular vision refers to the area seen with one eye.

notion of *orientation and attractiveness to a global attractor* can be extracted from the orientational movement of migratory birds. In accordance to [64], migrating birds use the magnetic field for direction finding, either towards magnetic poles (polewards, northern or southern) or the magnetic equator (equatorwards).

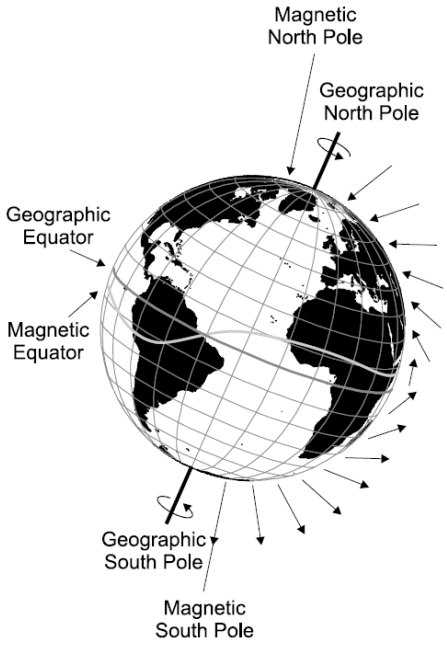


Figure 9: Schematic drawing of the Earth"s magnetic field (redrawn from Wiltschko and Wiltschko [15]). The arrows show the course of the magnetic field lines and their length is drawn relative to the magnetic field intensity at different latitudes. The magnetic field intensity is strongest at the magnetic poles (about 68 μT) and weakest at the magnetic equator (about $23\mu T$). The steepness of the magnetic field lines relative to the surface of the Earth shows the angle of inclination, which is maximal at the magnetic poles ($\pm 90^o$) and minimal (0^o) at the magnetic equator.

It is a biological fact that migrating birds can take advantage of the environment for their movements. Migrating birds can sense the Earth's magnetic field and orientate themselves with the ease of a compass needle. This ability is a massive boon for these birds, keeping frequent flyers on the straight and narrow. The use of magnetic

information in bird orientation was discussed in the past, when von Middendorff [65] (1859) proposed what someone, in modern terms, would call a 'magnetic compass', and Viguier [66] (1882) suggested that displaced pigeons use total magnetic intensity and inclination to determine their home direction. In accordance to [64], migrating birds use the magnetic field (Fig. 9) for direction finding, either towards magnetic poles (polewards, northern or southern) or the magnetic equator (equatorwards).

In addition, the findings summarized in another paper [15], clearly show that the geomagnetic field provides important orientational information[4] .

2.3 The Lotka-Volterra (LV) competition model

Often, biological systems are studied in terms of simple non linear mathematical models [39] which aim at modeling natural and biological processes using analytical techniques and tools. Population dynamics has traditionally been the dominant branch of mathematical biology which studies how populations of species change in time and space as well as the processes that cause these changes. Information about population dynamics is of fundamental importance for policy making and planning.

Population dynamics can be modeled with a simple balance equation that describes how the overall population size of a species changes from time to time as a result of species interaction with resources, competitors, mutualists and natural enemies. This study adopts similar ideas so as to design congestion avoidance strategies

[4] A recent study by Heyers et al. [67], strongly supports the hypothesis that migratory birds use their visual system to navigate using the magnetic field. In accordance with Heyers, the magnetic field or magnetic direction may be perceived as a dark or light spot which lies upon the normal visual field of the bird, and which, of course, changes when the bird turns its head.

for competing streams of data flows. The proposed approach is based on *a deterministic competition model which involves interactions among species that are able to coexist, in which the fitness of one species is influenced by the presence of other species that compete for at least one limiting resource.* An in depth investigation and modeling of competitive interactions between organisms provides an initial basis for predicting outcomes.

This study focuses on the Lotka-Volterra (LV) competition model [33], [34], which is considered to be one of the most studied mathematical models of population biology. The proposed LV-based congestion control approach is based on one of the basic characteristics of the Lotka-Volterra competition model. As thoroughly discussed below, the LV competition model guarantees that the only non-negative equilibrium point of the LV-based system ensures coexistence of all flows, stability, and fairness among active flows when a few conditions are satisfied.

2.3.1 The microscopic level of system behavior: Interactions and rules

From the *microscopic level*, the species within an LV-based ecosystem follow a set of rules:

1. The population of each species i grows at a given intrinsic growth rate r_i in the absence of all other competing species.

2. Each species i reproduces proportionally to the population of the same species i, by an intra-specific competition coefficient β_i. The parameter β_i represents the competitive effects among individuals of species i.

3. Each species i reproduces proportionally to the population of species j, by an inter-specific competition coefficient α_{ij}. The parameter α_{ij} represents the competitive effects of species j on growth of species i.

If $\alpha_{ij} < \beta_i$, then the competitive effect of species j on population growth of species i is less than that of an individual of species i. The rules can be described in an aggregated form by a set of differential equations.

2.3.2 The mathematical model

The generalized form of an n-species LV system is expressed by a system of ordinary differential equations:

$$\frac{dx_i}{dt} = x_i \left(r_i - \frac{\beta_i r_i}{K_i} x_i - \frac{r_i}{K_i} \left(\sum_{j=1, j \neq i}^{n} \alpha_{ij} x_j \right) \right), \tag{4}$$

for $i = 1, ..., n$, where $x_i(t)$ is the population size of species i at time $t\,(x_i(0) > 0)$. Also K_i is the carrying capacity of species i. K_i is the maximum number of individuals of species i that can be sustained by the biotope in the absence of all other species competing for the same resource when $\beta = 1$. Otherwise, the maximum population size of species i can reach K_i/β_i. If only one resource exists and all species (having the same carrying capacity K) compete for it, then K can be seen as the resource's capacity. The LVCC approach is developed on the basis of the n-species LV model, assuming that the n species have the same characteristics: $r_i = r$, $K_i = K$, $\beta_i = \beta$ and $\alpha_{ij} = \alpha$ for every i, j.

2.3.3 The macroscopic level of system behavior

In the *macroscopic level*, the system emergent behavior is described in terms of the species populations evolution. According to the analytical study of the LV competition model by Brauer and Chavez [39], the system can be in one of the following states:

1. All species in the system can survive and coexist.

2. At least one species can survive, out-competing the rest and condemning them to extinction.

It is beyond any doubt that these states cannot be intuitively deduced from the microscopic rules.

The system of Eq. 4 cannot be analytically solved, but some information about the behavior of its solutions can be obtained. Instead of trying to solve for $x_1, x_2, ..., x_n$ as functions of t, t can be eliminated so as to look for the relation between all x_i, for $i = 1, ..., n$. In geometric terms, the *phase plane*, the (x,y) plane is a useful visual aid, which can be used to predict the outcome of competition over time (emergent system behavior). The dynamic path of this many-commodity ecosystem can be determined on the basis of the trajectories (or orbits) of solutions [5] . Conceptually, the state of such a system is represented by a point (a vector of n components) lying on an n-dimensional trajectory, and the movement of this point over the trajectory through time is governed by the system of differential equations.

[5]Curves in the plane representing the functional relationship between all x_i, $i = 1, ..., n$, with the time t as parameter.

Unfortunately, most problems that arise in the real world are not linear, and in most cases, nonlinear systems can not be solved - there is typically no method for deriving a solution to the equations. When confronted with a nonlinear differential equation system such as the n-dimensional system of Eq. 4, an approximate solution may be satisfactory. One method to find approximate solutions is *linearization* near a constant solution. The equilibria are constant solutions of the system of differential equations (e.g. Eq. 4) that satisfy the equations $dx_i/dt = 0$, for $i = 1, ..., n$. Geometrically, an equilibrium is a point in the phase plane that is the orbit of a constant solution. Therefore, it seems more convenient to study the behavior of the system around equilibria (if any) and investigate their stability.

In this study, the 2-species LV competition model is taken into consideration when investigating the stability of equilibria. The system of differential equations is given by:

$$\frac{dx_1}{dt} = f(x_1, x_2) = rx_1\left(1 - \frac{\beta}{K}x_1 - \frac{\alpha}{K}x_2\right), \quad \frac{dx_2}{dt} = g(x_1, x_2) = rx_2\left(1 - \frac{\beta}{K}x_2 - \frac{\alpha}{K}x_1\right). \tag{5}$$

An equilibrium is a solution (x_1^*, x_2^*) of the pair of equations $dx_1/dt = f(x_1^*, x_2^*) = 0$, $dx_2/dt = g(x_1^*, x_2^*) = 0$. Four possible equilibria can be distinguished: (a) $(0, 0)$, (b) $\left(\frac{K}{\beta}, 0\right)$ with $\frac{K}{\beta} > 0$ and $f(\frac{K}{\beta}, 0) = 0$, (c) $\left(0, \frac{K}{\beta}\right)$ with $\frac{K}{\beta} > 0$ and $g(0, \frac{K}{\beta}) = 0$, and (d) $\left(\frac{K}{\alpha+\beta}, \frac{K}{\alpha+\beta}\right)$ with $\frac{K}{\alpha+\beta} > 0$, $f\left(\frac{K}{\alpha+\beta}, \frac{K}{\alpha+\beta}\right) = 0$ and $g\left(\frac{K}{\alpha+\beta}, \frac{K}{\alpha+\beta}\right) = 0$.

The first equilibrium $(0, 0)$ describes the situation in which no species survive. Both equilibria $\left(\frac{K}{\beta}, 0\right)$ and $\left(0, \frac{K}{\beta}\right)$ describe the situation in which one species survives but the other species loses the struggle for existence and becomes extinct. The

last equilibrium describes the coexistence of species. In the presence of n species, there will be n equilibrium points: all species coexist, all species extinct, and $n-2$ combinations of some species surviving and some vanishing. As from a biological point of view only non-negative population sizes are of interest; only equilibria having non-negative coordinates are considered.

The behavior of solutions near an equilibrium can be determined by the behavior of solutions of the linearization at the equilibrium. The linearization of the system 5 using Taylor's theorem at the equilibrium (x_1^*, x_2^*) (neglecting higher order terms) gives the coefficient matrix:

$$M\left(x_1^*, x_2^*\right) = \begin{pmatrix} f_x\left(x_1^*, x_2^*\right) & f_y\left(x_1^*, x_2^*\right) \\ g_x\left(x_1^*, x_2^*\right) & g_y\left(x_1^*, x_2^*\right) \end{pmatrix} = \frac{r}{K} \begin{pmatrix} K - 2\beta x_1^* - \alpha x_2^* & -\alpha x_1^* \\ -\alpha x_2^* & K - 2\beta x_2^* - \alpha x_1^* \end{pmatrix}. \tag{6}$$

which is called the *community matrix* of the system at the equilibrium (x_1^*, x_2^*). It describes the effect of the size of each species on the growth rate of itself and the other species at equilibrium.

An equilibrium (x_1^*, x_2^*) is said to be *stable* if every solution $(x_1(t), x_2(t))$ with $(x_1(0), x_2(0))$ sufficiently close to the equilibrium remains close to the equilibrium for all $t \geq 0$. An equilibrium (x_1^*, x_2^*) is said to be *asymptotically stable* if it is stable and if, in addition, all solutions with $(x_1(0), x_2(0))$ sufficiently close to the equilibrium tend to the equilibrium as $t \to \infty$.

The asymptotic stability or instability of a linear system can be determined by the eigenvalues of the matrix M. The sum of the eigenvalues is the *trace* of the matrix M and the product of the eigenvalues is the *determinant* of the matrix M. An

equilibrium (x_1^*, x_2^*) is asymptotically stable if all eigenvalues of the coefficient matrix of the linearization at this equilibrium, M, have negative real part, specifically if:

$$\text{tr } M\left(x_1^*, x_2^*\right) = f_x\left(x_1^*, x_2^*\right) + g_y\left(x_1^*, x_2^*\right) < 0, \tag{7}$$

$$\det C\left(x_1^*, x_2^*\right) = f_x\left(x_1^*, x_2^*\right) g_y\left(x_1^*, x_2^*\right) - f_y\left(x_1^*, x_2^*\right) g_x\left(x_1^*, x_2^*\right) > 0. \tag{8}$$

Based on Eqs. 7 and 8, the stability of the equilibrium points depends on the relationship between α and β. When $\beta > \alpha$, only the equilibrium $\left(\frac{K}{\alpha+\beta}, \frac{K}{\alpha+\beta}\right)$ is asymptotically stable (having positive determinant and negative trace), while the other three equilibria of the system 5, namely $(0,0)$, $\left(\frac{K}{\beta}, 0\right)$ and $\left(0, \frac{K}{\beta}\right)$ are classified as unstable. On the other hand, when $\beta < \alpha$, only the equilibria $\left(\frac{K}{\beta}, 0\right)$ and $\left(0, \frac{K}{\beta}\right)$ are asymptotically stable.

Each linearization provides a good approximation to the behavior of system 5 near the corresponding equilibrium point. In addition, the eigenvalues at each equilibrium determine the stability of the equilibrium. However, the local linearizations do not tell us what is happening in the phase plane far from the equilibria. In order to provide a better understanding of the system behavior far from equilibria under different initial conditions *phase portraits*[6] graphs are considered. For each species, there is a straight line on the phase portrait called a zero isocline[7] (or nullcline). Any given point along, for example, species 1's zero isocline represents a combination of populations of the two species where the species 1 population does

[6] A phase portrait is a geometric representation of a dynamical system which depicts the system's trajectories, the stable steady states and the unstable steady states in the phase plane.

[7] The zero isocline for a species is calculated by setting the growth rate, dx/dt, equal to zero and solving for x.

not increase or decrease. Figure 10 shows the zero isoclines for species 1 (left) and species 2 (right).

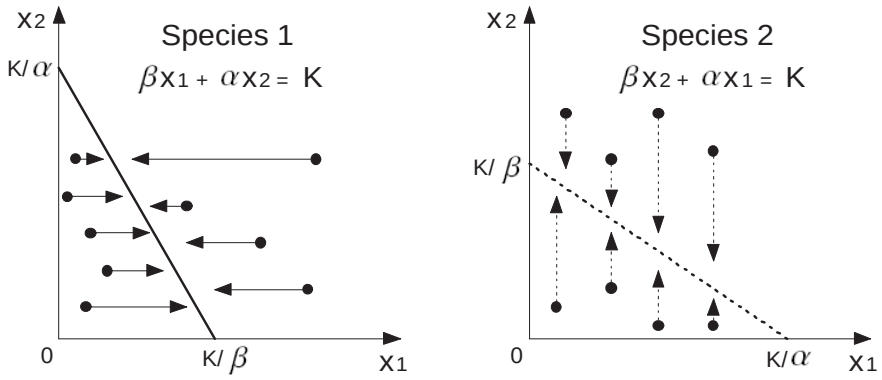


Figure 10: Zero isoclines for (a) species 1 and (b) species 2.

Note that the zero isoclines divide each phase portrait into two parts. Below and to the left of the isocline the population size increases because the combined populations of both species are less than the maximum population size of each species (K/β, while above and to the right the population size decreases because the combined populations are greater than the K/β. For the phase portrait of species 1, the isocline intersects the graph on the x-axis when x_1 reaches K/β and no individuals of species 2 are present. The isocline intersects the phase portrait on the y-axis at K/α, when the carrying capacity of species 1 is filled by the equivalent number of individuals of species 2 and no individuals of species 1 are present. The intersections of the isocline for species 2 are essentially the same, but on different axes.

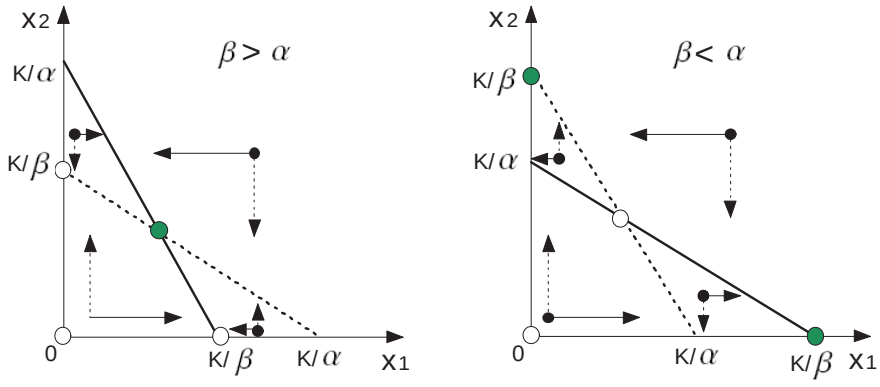


Figure 11: Zero isoclines for species 1 and 2 when (a) $\beta > \alpha$ and (b) $\beta < \alpha$.

The phase portraits of Fig. 11 include both species' isoclines, and illustrate the possible outcomes of interspecific competition depending on where each species' isocline lies in relation to the other. In each phase portrait, the solid line represents the isocline of species 1, and the dashed line represents the isocline of species 2. The arrows indicate the trajectories of species' population sizes for different initial values while converging/diverging to/from systems' equilibria.

Fig. 11(a) shows the isoclines of both species when $\beta > \alpha$. It can be observed that regardless of the initial values of population sizes, all trajectories head toward the intersection of isoclines. More specifically, below and above the two isoclines both populations increase and decrease respectively. Also, when the species populations are between the isoclines their trajectories always are directed to the intersection point. The coordinates of the intersection point are $\left(\frac{K}{\alpha+\beta}, \frac{K}{\alpha+\beta}\right)$ and correspond to the equilibrium point which was analytically found to be stable when $\beta > \alpha$. Fig. 11(a) verifies this finding graphically, according to which the two species, rather

than outcompeting one another, are able to coexist at this asymptotically stable equilibrium point (shaded circle). The other three equilibrium points represented by open circles are unstable as no trajectory tends to $(0,0)$, $\left(\frac{K}{\beta},0\right)$ and $\left(0,\frac{K}{\beta}\right)$.

Fig. 11(b) illustrates the phase portrait of the system when $\beta < \alpha$. As analytically found earlier, the equilibria $\left(\frac{K}{\beta},0\right)$ and $\left(0,\frac{K}{\beta}\right)$ are both asymptotically stable (shaded circles). These equilibria divide the phase portrait into two regions, one containing initial points for which trajectories tend to $\left(\frac{K}{\beta},0\right)$, and the other containing initial points for which trajectories tend to $\left(0,\frac{K}{\beta}\right)$. In this case, coexistence of of the two species is impossible, and one species will always win the competition for survival (i.e. competitive exclusion of one species by the other species). The winner of this competition is determined by the initial population sizes.

2.3.4 Stability analysis of the generalized Lotka-Volterra model

The existence of globally stable equilibria in the generalized Lotka-Volterra ecosystem has been investigated extensively in literature [68], [69]. Prior to investigating stability issues, Eq. 4 is expressed in vector form:

$$dx_i/dt = X\left(b - Ax\right), \tag{9}$$

where $x = (x_1, ..., x_n)^T$ is an n-dimensional state vector, $X = diag\,(x_1, ..., x_n)$ is an $n \times n$ diagonal matrix, $b = (b_1, ..., b_n)^T$ is an n-dimensional real vector, and $A_{ij} = (a_{ij})$ is an $n \times n$ community matrix.

The n-dimensional Euclidean space is denoted by R^n. Let I be a subset of $N = \{1, ..., n\}$ such that $x_i^* = 0\ \forall i \in I$, where $x^* = (x_1^*, ..., x_n^*)^T$ is a non-negative

equilibrium point of Eq. 4. If we let $J = N - I$ such that $x_j^* > 0 \ \forall j \in J$, we can define $R_I^n = \{x | x \in R^n, x_i \geq 0 \text{ for } i \in I \text{ and } x_j > 0 \text{ for } j \in J\}$, as the set corresponding to x^*. Below, two definitions and a theorem related to the concept of stability are provided:

Definition 1. When A is an $n \times n$ real matrix, $A \in S_W$ implies that there exists an f $n \times n$ positive definite diagonal matrix W such that $WA + A^T W$ is positive definite.

Theorem 1. A real, symmetric matrix A is positive definite if and only if all its eigenvalues are positive.

Definition 2. A non-negative equilibrium point x^* of Eq. 4 is called asymptotically stable with respect to the set R_I^n, iff: (a) The equilibrium point $x^* \geq 0$ is stable with respect to R_I^n, namely, if for every $\epsilon > 0$ there exists $\delta(\epsilon)$ such that if $|x^0 - x^*| < \delta(\epsilon)$ and the solution $x(t) \in R_I^n$, then $|x(t) - x^*| < \epsilon$ for $t \geq 0$, and (b) every solution converges to x^* as $t \to +\infty$, if $x^0 \in R_I^n$.

According to Takeuchi and Adachi [68], if $A \in S_W$, then Eq. 9 has a non-negative and stable (in the sense of Definition 2) equilibrium point for every $b \in R^n$. This stability concept is being investigated in the proposed model assuming that the n species have the same characteristics. That means that in terms of Eq. 4, we have $r_i = r$, $K_i = K$, $\beta_i = \beta$ and $\alpha_{ij} = \alpha$ for every i, j. Based on these assumptions, the

matrixes A and b of Eq. 9 are constructed as follows:

$$A = \begin{pmatrix} \frac{\beta r}{K} & \frac{\alpha r}{K} & \cdots & \frac{\alpha r}{K} \\ \frac{\alpha r}{K} & \frac{\beta r}{K} & \cdots & \frac{\alpha r}{K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha r}{K} & \frac{\alpha r}{K} & \cdots & \frac{\beta r}{K} \end{pmatrix} = \frac{r}{K} \begin{pmatrix} \beta & \alpha & \cdots & \alpha \\ \alpha & \beta & \cdots & \alpha \\ \vdots & \vdots & \ddots & \vdots \\ \alpha & \alpha & \cdots & \beta \end{pmatrix} = A^T, \quad b = (r, r, ..., r)^T. \quad (10)$$

Based on Definition 1, if $A + A^T$ is positive definite then $A \in S_W$ which means that Eq. 9 has a non-negative and stable equilibrium point. As illustrated above, $A + A^T = 2A$ is a symmetric real matrix which can be considered to be positive definite if and only if all its eigenvalues are positive (by Theorem 1). Using the characteristic polynomial of the $n \times n$ matrix $A + A^T$ all its eigenvalues can be computed which are expressed by: $\lambda_1 = \beta + (n-1)\alpha$, and $\lambda_i = \beta - \alpha$ for $i = 2, .., n$. It can be easily observed that all the eigenvalues are positive if $\beta > \alpha$ and $\alpha, \beta > 0$. Therefore, *the proposed system has a non-negative and stable equilibrium point when inter-specific competition is weaker than intra-specific competition.*

Intuitively, the next step is to seek the main attraction point of the n-dimensional LV system. As demonstrated in [68], Eq. 9 can be transformed to a linear complementarity problem (LCP) (q, M) [70] whose feasible solution can be obtained using algorithms found in literature like PATH [71]. Hence, the non-negative stable equilibrium point of the proposed system obtained using the using PATH algorithm [71] is given by:

$$x_i^* = \frac{K}{\alpha(n-1) + \beta}, \; i = 1, ..., n. \quad (11)$$

Eq. 11 reveals that the main attraction point is a coexistence solution. So, *when the condition $\beta > \alpha$ is satisfied, all species coexist (survive).*

The aforementioned analysis is used in Chapter 4 for developing the Lotka-Volterra-based congestion control approach (LVCC).

2.4 Related Work

2.4.1 Congestion control approaches in WSNs

Early studies in the area of sensor networks had mainly focused on more fundamental networking problems, e.g. medium access control (MAC), topology, routing, and energy efficiency, and had largely ignored network performance assurances. Lately, with the emergence of mission-critical applications (e.g. health monitoring), there has been increased interest in performance control mechanisms, so as to avoid congestion caused by the uncontrolled use of scarce network resources.

Various CC approaches can be found in WSNs literature based on traffic manipulation (e.g. rate adaptation to network changes [23], [21], [22], [47] multi-path routing [45], [72], [42]), topology control (e.g. clustering formation [73]), and network resource management (e.g. power control, multiple radio interfaces [46]). The majority of CC approaches are based on rate control that alleviates congestion by throttling the injection of traffic in the network. However, rate control attempts to decrease nodes' reporting rate during transient or persistent congestion phenomena and may result in the deterioration of the offered quality of service, perhaps when

needed the most. In addition, rate control can not be efficiently used during transient congestion phenomena caused by aperiodic and short term packet bursts (e.g. in event monitoring applications) due to the slowness of rate-based approaches to react. Also, approaches that require the sink to regulate the sensors' sending rates [23] seem quite unrealistic, especially in large-scale sensor networks or in networks deployed in an area with obstruction characteristics. Furthermore, clustering formation assumes special roles in the network (e.g. clusterheads), while additional mechanisms are needed for maintaining and re-assigning roles. Also, areas around clusterheads may progressively become collision hot spots. Congestion mitigation based on power control and multiple radio interfaces seems unrealistic in WSNs since the low-cost nodes incentive is violated. On the other hand, multi-path routing has potential to effectively and efficiently alleviate congestion without deteriorating the offered network QoS. CAR [72] dynamically discovers a congestion zone and routes high priority packets inside the zone while low priority packets are routed outside the zone. One of the problems with this approach is how to categorize traffic. Biased Geographical Routing (BGR) [45] alleviates congestion by splitting traffic flows and performing rate control. Both studies in [45], [72] use location information provided by the Global Positioning System (GPS) to discover congestion regions. However, GPS can work only outdoors in the absence of any obstruction, while GPS receivers are expensive and not suitable in the construction of small cheap sensor nodes. In a similar approach, TADR [42], a mixed potential field is constructed using depth and normalized queue length to route packets around the congestion areas and scatter

the excessive packets along multiple paths consisting of idle and under-loaded nodes. However, the dynamic conditions of the wireless medium which may cause excessive packet loss in WSNs are not considered, while the authors evaluate their scheme assuming a perfect (but practically infeasible) MAC protocol that provides a stable radio link without causing collisions.

Some of the aforementioned CC schemes [21], [45] are based on traditional methodologies and protocols known from the Internet, for example, the AIMD rate control. Due to its ACK-controlled packet injection method, AIMD does not support loss differentiation in order to distinguish wireless losses from congestion losses. Thus, it is shown to provide unsatisfactory performance in wireless environments where high packet loss rates are often attributed to the time-varying conditions of the wireless channel, e.g. interference, multi-path fading, etc. In addition, AIMD takes a very long time to converge or recover from a burst of loss something which is quite usual in event-based wireless sensor networks. The problem of delay is worsened when the ACKs are sent end-to-end between sensor nodes and the sink (as done in both [21] and [45]).

AIMD policy is not very effective in WSNs because it results in a saw-tooth rate behavior which may violate the QoS requirements (e.g. fidelity of the reported events) and it is inefficient. More specifically, the AIMD policy exhibits a linear growth of the number of packets sent, combined to an exponential reduction when a congestion occurs. The result is a saw tooth behavior representing the probe for bandwidth which makes it inappropriate for real time traffic.

Conventional optimization techniques for CC may also be adopted, but these rely on complex mathematical models and have proven hard to apply in the context of autonomous decentralized environments, while the sensitivity of these models to the dynamic and unpredictable environment is a challenge. Also, centralized multi-objective optimization is not practical especially in the context of large-scale autonomous networks.

2.4.2 Swarm intelligence approaches for solving network problems

From the viewpoint of swarm intelligence (SI), a considerable number of models based on self-propelled particles have been developed to solve a variety of problems by means of collective motion. The research framework behind the majority of network-oriented studies involving self-propelled particles was fueled by the ACO theory proposed by Dorigo et al. [74]. Both theories were successfully involved in network-oriented studies, especially in the field of ad-hoc and mobile ad-hoc networks (MANETs). A large number of SI-based routing approaches for MANETs can be found in [75], while some notable examples are presented below.

Driven by the collective behavior of ants in finding paths from the colony to food, many researchers [12], [76], and [51], developed ACO-based algorithms to solve the problem of routing in ad-hoc networks. In ACO, artificial ants build solutions by moving on a graph of nodes and, by mimicking real ants, deposit artificial pheromone on the links (that form the route traveled by each ant) in such a way that future artificial ants can build new better solutions. The work of Di Caro et

al. [12] incorporates congestion awareness in an end-to-end manner (AntHocNet). According to [77], AntHocNet outperformed the Ad-hoc On-demand Distance Vector (AODV [78][8]) routing protocol in terms of delivery ratio, end-to-end delay and delay variation (jitter). An important observation was that the advantage of AntHocNet over AODV grew for larger networks, especially in terms of overhead, suggesting that AntHocNet is more scalable than AODV. Rajagopalan et al. [76] presented an Ad-hoc Networking with Swarm Intelligence (ANSI) routing protocol with congestion-aware characteristics. The ANSI protocol was found to outperform AODV in terms of packet delivery, number of packets sent, end-to-end delay, and jitter experiencing fewer route errors as compared to AODV. Xiangquan et al. [51] developed an ACO algorithm to optimize energy, congestion and load balancing along with providing efficient routing in ad-hoc networks.

Based on the aforementioned results, it seems that nature-inspired approaches are able to outperform conventional routing algorithms in (mobile) ad-hoc networks. However, the aforementioned approaches are not well suited for the unique features and application requirements of WSNs. According to Akyildiz et al. [16] there are some important differences between ad-hoc and sensor networks. For example, sensor nodes are limited in power, computational capacities, and memory whereas nodes involved in ad-hoc networks may be more powerful and less constrained machines like laptops and PDAs. Also, the number of sensor nodes in a sensor network can be several orders of magnitude higher than the nodes in an ad-hoc network, and can often be densely deployed over large areas. Sensor nodes are prone to failures

[8]AODV [78] is a conventional single-path ad-hoc routing protocol.

and are mostly statistically deployed. On the other hand, ad-hoc nodes are usually less densely deployed in small areas.

These differences drive the necessity for novel CC strategies for WSNs, on which this study focuses. It is worth pointing out that there are not many nature-inspired studies that have explicitly focused on avoiding or combating congestion in sensor networks. A notable related approach is the ant-based multi-QoS routing protocol for sensor networks (AntSensNet) [13]. AntSensNet combines a hierarchical structure of the network with the principles of ACO-based routing, whilst trying to satisfy the QoS requirements of different kinds of traffic requested by the applications. By using clustering, AntSensNet aims to avoid congestion after quickly judging the average queue length and solving convergence problems, which are typical in ACO. However, AntSensNet has some important drawbacks (e.g. computation and communication overhead and large number of tunable parameters) that increase protocol's complexity and decrease the protocol responsiveness to changes (e.g. node failures) in the network. These are discussed at the end of the performance evaluation section when comparing AntSensNet to Flock-CC. **The potential of the Flock-CC approach emerges from its simplicity (a set of simple rules, two tunable parameter) and its fast responsiveness to network and traffic changes since only local information are used in the decision-making process**.

2.4.3 Population biology approaches for congestion control

Previous work on congestion control involving mathematical models of population biology is basically applicable to the Internet on the basis of either providing a new way of thinking and combating congestion [37], [79] or improving the current TCP congestion control mechanism [80], [81], [82].

Competition theory can serve as a basis for mapping Internet congestion control. Recent work by Iguchi et al. [79] and Hasegawa and Murata [37] focused on a new TCP congestion control mechanism (TCP Symbiosis) based on the Lotka-Volterra competition model, that utilized available bandwidth information obtained from inline measurement techniques. The aforementioned model which describes changes in the population of species was applied to the congestion window updating mechanism of TCP. This application can be done by considering the number of a single species as window size of a TCP connection, a carrying capacity as a bottleneck link bandwidth, and competition among species as a bandwidth share among competitive TCP connections. Each source was in charge of adjusting its congestion window based on competitive interactions with other flows sharing the available network bandwidth. This evaluation required each source to be aware of the data transmission rates of all other competing flows, something which is quite unrealistic in the current Internet. However, this quantity can be approximated by each source via subtracting the estimated available network bandwidth from the physical bandwidth of the bottleneck link. Authors state that the TCP Symbiosis

approach exhibits remarkable results in terms of stability, convergence speed, fairness and scalability. However, this approach depends heavily on measurement tools that measure the physical capacity and available bandwidth of end-to-end network paths. A few measurement tools have been proposed and evaluated in simulation and over a limited number of Internet paths, but there is still great uncertainty in the performance of these tools over the Internet at large [83]. Errors and uncertainties that may emerge depend mainly on on hardware and software specifications (CPU frequency, OS version) and resulting in latency in packet sending operations [84].

In addition to Lotka-Volterra competition model, the Lotka-Volterra prey-predator model can be used for controlling congestion. A multidisciplinary conceptual framework providing principles for designing and analyzing bio-inspired Internet congestion control algorithms was proposed by Analoui and Jamali [80]. A bio-inspired congestion control (BICC) mechanism was designed and implemented in the context of the proposed framework taking into account that the relations between entities involved in Internet congestion control mechanisms (i.e., routers, links and hosts) are similar to population interactions like predator-prey. More specifically, the Internet was viewed as an ecosystem that connects a wide variety of habitats such as routers, hosts, links, operating systems, etc. Some species such as, congestion window, packet drop, queue length, and link utilization were considered to live those habitats. The population size of each species is affected as a result of inter-species interactions but it should be controlled so as to avoid congestion. Towards this

direction, the predator-prey behavior was employed to regulate the population size of each species, thus contributing to the elimination of congestion in the Internet. BICC is a unified source/AQM (Active Queue Management) approach that addresses both the source and the AQM algorithms. This approach was extended in [81] and [82] which attempt to examine how the Lotka-Volterra predator-prey model can be applied to the dynamics of the Internet while addressing stability, robustness, scalability, and fairness. In particular, BICC involves two predator species living in congested routers which can control the population size of a single prey species. Congestion was quantified by the population sizes of these species which are evaluated at each congested router taking into account the incoming packet rate as well as the local queue size. In this way, every router infers the level of congestion and uses a marking probability function to generate explicit congestion notification signals. Therefore, each source is informed of congestion levels and adjusts its sending rate accordingly. According to the authors, BICC exhibits fairness, acceptable network throughput and utilization but results show slow adaptation to traffic demand (long convergence time).

The approaches discussed above are based on the end-to-end model of the Internet, which is completely different from the hop-by-hop nature of autonomous decentralized networks like WSNs. **The novelty of the proposed Lotka-Volterra Congestion Control (LVCC) approach presented in Chapter 4 lies in the fact that the LV model is applied to WSNs in a hop-by-hop manner**.

2.5 Concluding remarks

This chapter provides an outline on the problem of congestion in WSNs focusing on the different types of congestion as well as on strategies needed for avoiding or mitigating congestion. In addition, this chapter gives an introduction to swarm intelligence with emphasis on the basic characteristics of the flocking behavior of birds, followed by an introduction to the Lotka Volterra competition model with emphasis on the stability analysis of the model. Finally, the chapter presents related studies for avoiding and controlling congestion is WSNs based on both conventional (Internet-like) and nature-inspired strategies and highlights the novelties that differentiate the two approaches proposed (Flock-CC and LVCC), presented in Chapter 3 and Chapter 4 respectively, from the previous work.

Chapter 3

The Flock-based congestion control (Flock-CC) approach

This section, firstly, presents the concept behind the Flock-CC model, explaining how the movement of packets is modeled as a flock of birds. Secondly, the elements of the Flock-CC model are defined on the basis of the characteristics of natural bird flocks. Thirdly, the elements of the model are composed together to form the Flock-CC protocol.

3.1 The concept

A WSN is viewed as a virtual ecosystem, where multiple packets are generated at source nodes and must be directed towards a dedicated sink node. The main idea of the proposed Flock-CC model is to *'guide' packets to form groups or flocks, and flow towards a global attractor (sink), whilst trying to avoid obstacles such as*

congestion regions and dead zones (regions with failing nodes) as illustrated in Fig. 12.

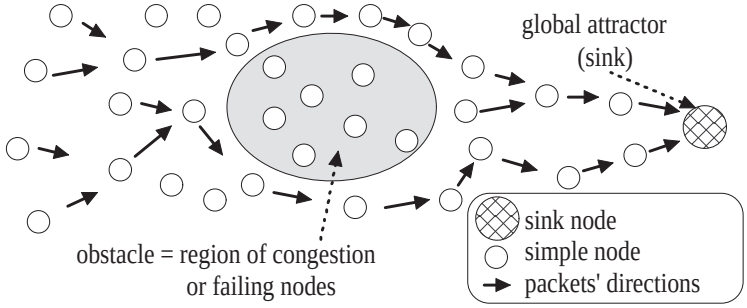


Figure 12: Packet flock moving towards sink whilst avoiding 'obstacles'.

In the Flock-CC model, each packet is analogous to a bird with dynamic position and direction updates, which 'flies' over the network undergoing successive hop-by-hop transitions over discrete points in the 2D space, defined by the positions of hosting nodes. The set of sensor nodes comprises the environment where packets move. The sequence of transitions determines the packet's (bird's) trajectory from its source to the sink. Note that this model differs from Couzin model, which is in a continuous 3D space.

Conceptually, in order to make moving packets behave like a flock, the basic characteristics of natural flocks discussed in the previous section are incorporated in the Flock-CC model. At each hop, a packet interacts (attraction and repulsion forces) with other packets located on neighboring nodes in the packet's FoV. Also the packet experiences a 'magnetic' force in the direction of the sink (global attractor). More specifically, the proposed Flock-CC approach is governed by 4 simple rules. Each packet is:

- *Rule* 1: repelled from neighboring packets located on nodes at close distance (i.e one hop away from packet) within the packet's FoV,
- *Rule* 2: attracted to neighboring packets located on nodes at medium distance (i.e. two hops away from packet) within the FoV,
- *Rule* 3: oriented and attracted to the global attractor under the influence of the environmental magnetic field, and
- *Rule* 4: experience some perturbation that may help the packets to pick a random route (i.e. trading exploration versus exploitation).

More detailed discussion on the implementation of these features on the Flock-CC approach follows next.

3.2 The Flock-CC model elements

The basic elements of the Flock-CC approach are defined in this section: (a) the repulsion and attraction zones and forces, (b) the artificial magnetic field, (c) the field of view (FoV), (d) the desirability function, and (e) randomness. The desirability function synthesizes the first three elements and determines the direction of movement of each packet, whilst randomness allows for exploration.

It is worth noting here that a basic design objective was to keep the number of adjustable parameters as small as practical in order to reduce complexity in optimizing parameter values for different network environments and conditions.

Consider a network of N autonomous nodes, $N > 0$, that are able to generate packets. A finite queue is associated with each node, while the node's throughput is constrained by the wireless channel capacity. A packet i and its current hosting node $n \in \{1, .., N\}$ (i.e. packet i is residing in the queue of node n) are taken as points of reference in order to define and discuss repulsion and attraction zones, the magnetic field and the field of view. The position of node n determines the position of the hosted packet i.

Each node n maintains four one-dimensional tables: (a) an attraction table, s_n, (b) a repulsion table, q_n, (c) a transmission table, t_n, and (d) an adjacency table, A_n. Each of these tables contains different information for a set of neighboring nodes for which node n has an active wireless link to. An entry s_{nm} of the attraction table of node n contains information about node m. The same applies for all tables. The contents of each table are discussed later.

All quantities defined herein are regularly sampled at discrete time intervals of T seconds at each sensor node. Then, the values of these quantities are broadcasted periodically (every T seconds) to all neighboring nodes (within transmission range), using a dedicated control packet. These periodic control packets are also used to update information about the connectivity of neighboring nodes.

Below, the five basic elements of the Flock-CC approach are defined.

3.2.1 Repulsion and attraction zones and forces

The design of the Flock-CC model was primarily motivated by the neighborhood-based model of Couzin et al. [11]. As discussed earlier, the Flock-CC model is applied on a two dimensional topological (discrete) space defined by a graph of nodes. The edges in the graph indicate the existence of wireless connection between two nodes. Each packet is expected to move in a hop-by-hop manner, over a set of nodes along the path to the sink.

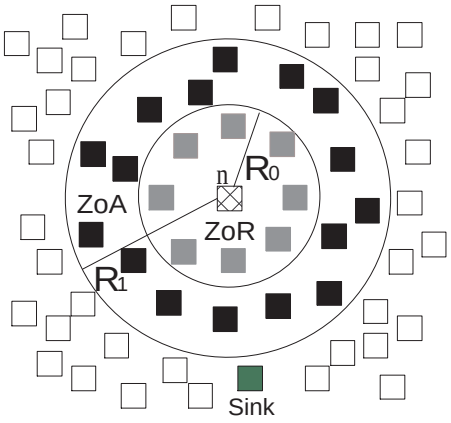


Figure 13: Representation of a sensor network. R_0 is the transmission range of node n and defines an area (zone of repulsion) that includes all packets located on grey-shaded nodes (one hop away). The outer area (zone of attraction) includes all packets located on black-shaded nodes (two hops away).

As shown in Fig. 13, the zone of repulsion is defined as a circle of radius R_0, the transmission range of node n, around packet i that includes all packets at close distance. Practically speaking, these packets reside in the queues of the grey-shaded nodes (practically 1 hop away). On the other hand, the zone of attraction is the outer zone of Fig. 13 and includes all packets at medium distance from packet i.

In practise, these packets reside in the queues of black-shaded nodes two hops away from node n. The zone of orientation proposed in Couzin model is not defined in the context of the Flock-CC approach since the notion of velocity matching (with neighbors in that zone) is not relevant in this problem's context.

The strength of the repulsive and attractive forces exhibited by packets on nodes one and two hops away respectively is selected to be proportional to the number of packets located on these nodes respectively. The number of packets on nodes one hop away (number of packets in the queue) is obtained directly through broadcasted control packets. These control packets are seen as a means of transferring knowledge (propagate information) within the environment (sensor network) that is observable by a packets' eyes.

On the other hand, in practise, the number of packets on nodes two hops away cannot be obtained directly through broadcasted control packets since (black-shaded) nodes two hops away are outside the transmission range of the current hosting node n. Thus, this number is locally inferred at node n by measuring the number of successfully transmitted packets sent from nodes one hop away to nodes two hops away. These packets are 'visible' due to the broadcasting nature of the wireless channel. Hence, packet i is repelled from packets residing in the queues of grey-shaded nodes and attracted to packets moving to black-shaded nodes.

3.2.2 Artificial magnetic field

In order to avoid unnecessary routing loops and to minimize packet losses, packets should be 'guided' to establish flight paths towards the sink. The artificial magnetic field is adopted for two reasons: a) to provide orientation and b) attraction of packets toward the sink.

The magnetic field of the Earth is responsible for guiding migratory birds to fly polewards or equatorwards as shown in Fig. 14(a). In the same context, the sink node is seen as an artificial magnetic pole within the sensor network and packets are expected to 'fly sinkwards' under the influence of the artificial magnetic field as shown in Fig. 14(b).

Orientation to the sink is deduced from the direction of the magnetic field (shorter paths point to the sink), while attraction to shortest paths to the sink is driven by the magnetic field strength. The magnetic field strength is higher through the shortest path to the sink and attenuates through longer paths. Orientation allows packets to perceive the direction of the sink, otherwise the absence of attraction to shortest paths towards the sink causes packets to wander around (trapped in loops between neighboring nodes). This looping behavior deteriorates the quality of service offered by the congestion control protocol. In addition, it is impossible for a packet to feel the attraction to shortest paths to the sink without orientating itself toward the sink.

In literature, there is a number of methods for implementing a magnetic field (or similarly called *gradient field*) in order to 'guide' packets in the direction of the

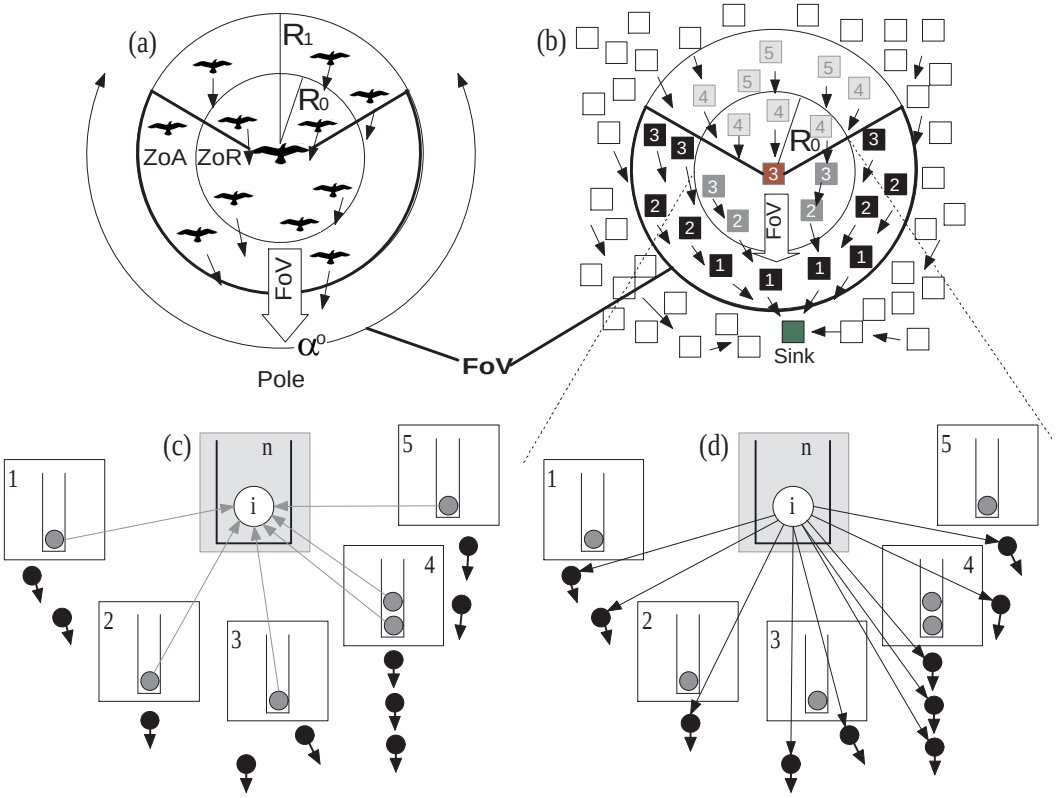


Figure 14: (a) A bird flock moving polewards under the influence of the magnetic field of the Earth (black arrows). The FoV of the bird placed in the center extends forward in the direction of the magnetic pole. (b) Packets generated in a sensor network will move sinkwards under the influence of the artificial magnetic field (black arrows). The number on each node indicates the hop distance from the sink (smaller numbers indicate closeness to the sink). The field of view (FoV) of packet i (on node n) extends forward in the direction of the sink. The ZoR and ZoA around packet i are redefined as 'circular' (in the sense of hop count) zones, except for an area behind the packet i that is outside the FoV. (c) Repulsion forces exercised on packet i from packets in the ZoR (heavy-gray-shaded packets). (d) Attraction forces exercised on packet i from packets in the ZoA (black-shaded packets).

sink as, for example, [85], [86], and [87], [88]. In these approaches, no routes are set prior to sending data, only costs are assigned to nodes. All the costs create a discrete gradient field whose minimum is located at the sink. When several sink nodes exist, several cost fields are determined. The cost field is either set up by an a-priori flooding stage (usually initiated from the sink) [86], [85] or on-demand with a request / response packet exchange [88]. The cost may take different forms such as hop distance, energy overhead, or even physical distance (in cases where a coordinate system like GPS exists). The gradient broadcast (GRAB) protocol [85] builds and maintains a gradient field, where the cost at each node is the minimum energy overhead to forward a packet from this node to the sink along a path[1] . Other routing protocols [86], [87] and [88] involve the hop distance of nodes to the sink as the cost of each node. The proposed Flock-CC approach uses the hop distance, which can be easily calculated, as the cost for building the magnetic field.

Orientation of packets to the sink is implemented as follows: The artificial magnetic field within the sensor network must point to the sink node. The direction of the sink is determined on the basis of the hop distance, $h_j(k)$, $j \in \{1, .., N\}$, indicating the number of hops between each node j and the sink at the kth sampling period. Each node evaluates its hop distance by adding one to the smallest hop distance of neighboring nodes. The sink node initiates the process of evaluating the hop distance value at each node by broadcasting (in a control packet) its zero

[1]GRAB protocol assumes that each node can estimate the cost of sending data to nearby neighbors (e.g., based on the signal-to-noise ratio of neighbors" transmissions).

hop distance. The hop distance value is gradually propagated (over control packets) from the sink (pole) to each node into the network (environment where packets live).

It is worth pointing out that the hop distance may change due to network topology modifications caused by node failures or displacement or other internal causes. If node n does not hear control messages from a given neighboring node for a maximum of T_{lost} seconds, then the node n assumes that the neighbor either has failed or gone out-of-range or is very busy. In this case, the unreachable neighbor is deleted from the adjacency table of node n containing the hop distances of all nodes one hop away. Similarly, the hop distance value of node n is re-evaluated if the loss of a neighboring node influences the value. Every change in hop distance values will be propagated backwards within the network and all affected nodes are expected to update their hop distance values. A simple hop distance protocol is adopted for this study. Note that it is anticipated that missing or incorrect hop count information will not affect Flock-CC adversely. This together with a detailed study of an optimized hop distance protocol are beyond the scope of the current study.

The magnetic field is visualized in Fig. 14(b), where the arrows point in the direction of the magnetic pole. A simpler way to visualize the magnetic field is to 'connect' the arrows to form 'magnetic field lines'. The magnetic field lines pass over paths of descending hop distance. Packets retrieve environmental information such as the direction of the artificial magnetic field (i.e. the hop distances of potential new hosting nodes) from their current hosting node.

Attraction to shortest paths toward the sink are implemented as follows: Besides orientation, the artificial magnetic field creates attraction to shortest paths to the sink. When a packet with orientation to the sink is ready to move, it has to choose between a number of potential new hosting nodes with different (equal or shorter) hop distances, taking into account the attraction and repulsion forces, as synthesized in the desirability function described later. The magnetic field guides packets to move to nodes closer to the sink where the strength of the field is higher. The movement of packets through nodes with shorter hop distances (forward movement) provides faster transitions to the sink and minimization of packet looping. This behavior is realized by enforcing differentiated attraction to potential new hosting nodes. In particular, packets perceive higher attraction to nodes closer to the sink (i.e having shorter hop distance) than to nodes at equal hop distance. A rule-based strategy for differentiated attraction is proposed for the Flock-CC protocol, as described in Section 3.3.

3.2.3 Field of view (FoV)

Motivated by the limited visual perception of birds, packet i cannot 'see' and interact with all packets on nodes in its neighborhood, as shown in Figs. 15(a) and (b). Packet i can perceive only a fraction of packets, i.e. those located in the FoV, on the observable world of the packet.

In general, the orientation of a bird's FoV can be set towards any direction, driving the movement of the bird accordingly. However, the orientation of a bird's

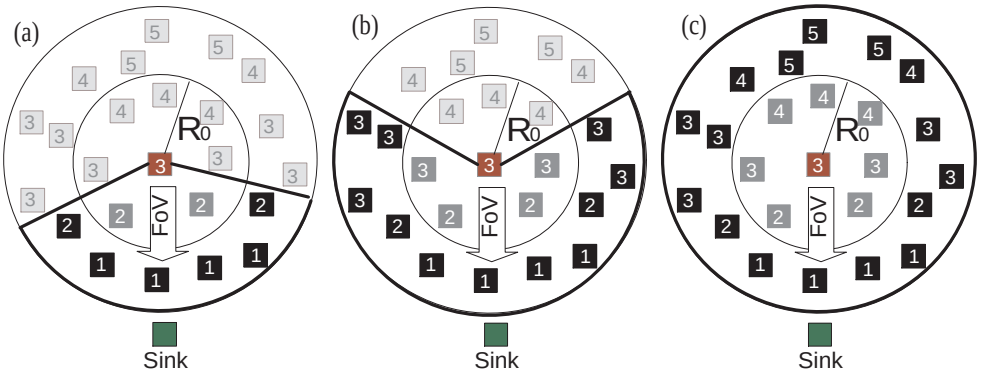


Figure 15: (a) Narrow FoV – includes nodes with shorter hop distance to the sink. (b) Wide FoV – includes nodes with shorter or equal hop distance to the sink. (c) Complete 360-degree FoV.

FoV can be affected by the presence of magnetic fields. Migratory birds that need to travel polewards turn the orientation of their head, and thus their FoV, towards the pole. In the same way, the FoV of packets is influenced by the artificial magnetic field and extends forward in the direction of the sink node as shown in Fig. 14(b).

The FoV of a packet may span from a few degrees up to 360^o. In the proposed approach, three different FoV apertures were considered as illustrated in Fig. 15. In the case of *narrow FoV*, packet i on node n (node in the middle) can 'see' only packets located on nodes with shorter hop distances than node n. In the case of *wide FoV*, packet i can 'see' packets located on nodes with shorter or equal hop distances to the sink compared to node n. In the extreme case, the 360-*degree FoV* of packet i includes all nodes in the neighborhood of the packet.

Simulations show that the complete 360-degree FoV, Fig. 15(c), allowing packets to move in any direction, exhibits poor performance [38]. The reason is the *absence of orientation, and thus attraction to the global attractor*. This caused high packet

concentration in the network, especially in the vicinity of source nodes, because packets become trapped in loops. Eventually, the overwhelming majority of packets were lost due to collisions in the wireless channel.

A natural way to address the problem of packet loops is to insist on using the narrow FoV where packets are allowed to move forward only, as shown in Fig. 15(a). This natural solution is too rigid, as it artificially excludes consideration of other nodes which are not necessarily on the direct path, but may have higher desirability, and makes paths towards the sink 'too narrow', thus increasing contention and node loading as well as minimizing the exploration of new paths. Furthermore, if the same nodes are always considered, then a higher power drain will be observed on these nodes and this may result in a shorter network lifetime. In addition, a problem may arise when front nodes either become unreachable (due to failures) or seen momentarily as unreachable (due to loss of control packets). In this case, packets reaching a dead end will be dropped.

Thus, the wide FoV of Fig. 15(b) that allows packets to be forwarded even to nodes that are placed at equal hop distance from the sink is selected in the proposed Flock-CC protocol. Each node n keeps the hop distances of its neighbors in its adjacency table, A_n.

Both the ZoR and the ZoA are redefined to involve the FoV as shown in Fig. 15. In this case, the ZoR is redefined as a circular zone of radius R_0, except for an area behind the packet i ('blind' area) that includes packets on nodes with longer hop distance than the current hosting node. The ZoR involves packets located in the

queues of the heavy-gray shaded nodes of Fig. 14(b). Therefore, packet i is repelled from these packets as illustrated in Fig. 14(c). Each node n keeps the queue sizes of each node in the ZoR in its repulsion table, q_n.

Similarly, the ZoA is the outer circular zone of Fig. 14(b) that includes packets on nodes at shorter or equal hop distance compared with the current hosting node (black-shaded nodes) within the FoV. As discussed earlier, since the number of packets on (black-shaded) nodes two hops away cannot be obtained with one broadcast message, each node n records the number of packets successfully transmitted from each node in the ZoR towards nodes in the ZoA in its attraction table, s_n. In addition, for each sampling period T, each node n records the total number of all packet transmission attempts (including retransmissions) from each node in the ZoR towards nodes in the ZoA in its transmission table, s'_n. The values of table s'_n are used for calculating the channel loading, as shown in the next section.

3.2.4 Desirability function

The repulsive and attractive forces (exercised by packets one and two hops away respectively) are synthesized by the decision making process which is invoked by the hosting node of each packet. The synthesis of repulsive and attractive forces is captured through a desirability function. The decision making process results in selecting the most desired next hop node on the basis of avoiding or minimizing congestion phenomena on next hop nodes. The desirability function is evaluated

once every sampling period and is used for directing each packet sent within this period to the next hop node, as discussed below.

An M-dimensional desirability vector, $\overrightarrow{D}(k)$, is used where $M \leq N$, is the number of potential new hosting nodes at the kth sampling period. The potential new hosting nodes are the nodes in the transmission range of the current hosting node n within the FoV. Each element, $D_{nm}(k)$, of the vector $\overrightarrow{D}_n(k)$ represents the desirability for each node $m, m \in \{1, .., M\}$ measured at node n. The desirability $D_{nm}(k)$ of every node m in the FoV is evaluated once every sampling period k (at the start of this period) and is used for directing each packet sent from its hosting node within this period to its next hop. The desirability of each node m evaluated at node n is given by:

$$D_{nm}(k) = s_{nm}^{norm}(k) - q_{nm}^{norm}(k), \tag{12}$$

where

$$s_{nm}^{norm}(k) = \begin{cases} \frac{s_{nm}(k)}{s'_{nm}(k)} & \text{if } s'_{nm}(k) > 0; \\ \xi & \text{otherwise,} \end{cases} \tag{13}$$

where $\xi \in [0, 1]$ is the *spreading variable* (discussed later), $s_{nm}(k)$ is the number of successfully transmitted packets from node m to nodes two hops away from node n, $s'_{nm}(k)$ is the total number of all packet transmission attempts at each node m, and

$$q_{nm}^{norm}(k) = \frac{q_{nm}(k)}{Q_m}, \tag{14}$$

where the function $q_{nm}^{norm}(k)$ is the number of packets in the queue of node m, $q_{nm}(k)$, divided by the queue capacity of each node m, Q_m.

The function $s_{nm}^{norm}(k)$ can be seen as a measure of quality of the wireless channel loading around node m as perceived from node n. The function $s_{nm}^{norm}(k)$ ranges from 0 to 1 and represents the normalized attraction force exercised on packet i by packets that moved (successfully transmitted) from each node m to nodes two hops away from packet i's current hosting node (i.e these packets are now within the ZoA). When $s_{nm}(k) \to 1$, the channel around node m is not congested and a large percentage of packets are successfully transmitted (few packet retransmissions are observed). As $s_{nm}(k) \to 0$, the channel is congested and a small percentage of packets are successfully transmitted, after a large number of retransmissions (i.e. the contention for this channel is very high).

An idle node m, i.e. with zero total transmission attempts ($s'_{nm}(k) = 0$) does not provide any evidence of the wireless channel quality in the vicinity of the node. Therefore, it cannot be said whether this node is either highly attractive, i.e. $s_{nm}^{norm}(k) = 1$, or highly repulsive, i.e. $s_{nm}^{norm}(k) = 0$. The spreading variable ξ is introduced as the normalized attraction force exercised by each node l that is idle ($s'_{nl}(k) = 0$). High values of ξ result in *packet spreading* since packets are attracted to idle nodes (most probably at the borders of the flock), whereas small ξ values lead to *coherent flock motion* (low spreading). These observations as well as further explorations of ξ values are presented in Chapter 5.

The function $q_{nm}^{norm}(k)$, with $0 \leq q_{nm}^{norm}(k) \leq 1$, is the percentage of queue occupancy at node m and represents the normalized repulsion force exercised on packet i by packets residing in the queue of each node m (i.e. these packets are now within

the ZoR). When $q_{nm}(k) \to 0$, the queue is empty or nearly empty, indicating low levels of congestion. On the other hand, as $q_{nm}(k) \to 1$, node m is considered congested due to high queue occupancy.

In the simple case, after evaluating the desirability of each node m, with $-1 \leq D_{nm}(k) \leq 1$, a packet can be forwarded from node n to node m^*, where the node m^* is a node within the FoV with the highest desirability. Even though the above approach allows for exploitation of existing good paths, it does not allow for exploration. Therefore to this basic approach we add the last flocking behavior characteristic, namely randomness.

3.2.5 Randomness

Randomness (or perturbation) is that part of nature which allows for exploration, and perhaps identification of better paths. Randomness also addresses the problem of always transiting to the same (highly desirable) new hosting nodes within a sampling period (recall that desirabilities are evaluated once every sampling period k), which can cause high queue occupancy on popular new hosting nodes. The next section includes a detailed description of how randomness is involved in the Flock-CC approach during the process of selecting a new hosting node.

3.3 The Flock-CC protocol

All the elements discussed above are composed together to form the Flock-CC protocol. Every time a packet is about to be sent, the decision making process

is invoked by the current hosting node to determine the new hosting node. The decision process employs three stages: (a) selection of direction (forward, sideways, backwards[2]) using the notion of the FoV and the magnetic fields, (b) sorting of all neighboring nodes in the selected direction in descending order by their desirability, and (c) probabilistic, biased (proportional to desirabilities) selection of the new hosting node.

The position of the sink plays an important role in evaluating the direction of movement of a packet. Under the influence of the magnetic field, a packet turns its FoV towards the sink and perceives the attraction of this artificial pole. The Flock-CC protocol uses a simple rule-based strategy of providing global attraction to the sink through differentiation of attraction to nodes in the FoV[3] . This strategy chooses the set of potential new hosting nodes among nodes placed in the FoV[3] using the following rules, given in decreasing order of priority:

1. Choose all nodes placed *closer* (smaller hop distance) to the sink. If there is at least one node among them with available buffer space then the packet moves forward.

2. If all nodes closer to the sink either have no available buffer space or are seen to be unreachable, choose all nodes placed at *equal hop distance* to the sink. If there is at least one node among them with available buffer space then the packet moves sideways.

[2]When a wide FoV is used, it is impossible for a packet to move backwards i.e. to nodes that are not observed by a packets's 'eye'. However, this restriction can be relaxed under the extremely rare situation where all front and side nodes are either unreachable or lack buffer space.

[3]Unless nodes within the the FoV are unavailable, e.g. failing or fully congested nodes.

3. Otherwise, the packet moves backwards.

The least priority rule allows a packet to move backwards to a node outside of the FoV in the extremely rare situation where all front and side nodes are either unreachable or lack buffer space.

The proposed rule-based strategy is quite similar to the rule-based process followed in Couzin's model [11] when determining the desired direction of each individual. This process was discussed in Chapter 2.

After choosing the set of potential new hosting nodes, the selected nodes are sorted in descending order based on their desirabilities. To implement randomness, the new hosting node is selected using a probabilistic selection. In order to provide a fair way of selection, the probability of selecting a node can be made proportional to either the desirability of that node, or the rank of the desirability of that node in the descending order of desirabilities.

In the Flock-CC approach, randomness is invoked whenever a packet is ready to choose its new hosting node. Randomness can be implemented on the basis of selection methods used in genetic algorithms [89], for example, roulette wheel selection or rank based selection. Selection methods are used for randomly selecting members from the population of chromosomes with probability proportional to their fitness.

The behavior of the Flock-CC protocol was tested for both roulette wheel selection and rank based selection. Results showed that the rank based selection prevailed over roulette wheel selection. Thus, rank based selection was adopted in the final

model of the Flock-CC protocol. Rank based selection ranks each individual i based on its fitness, f_i, and a new fitness value, f'_i, is assigned according to the rank the individual receives. *Individuals in Flock-CC are the potential new hosting nodes, M, and desirability is seen as the fitness of each potential new hosting node.* The weakest (in terms of desirability) potential new hosting node receives a new fitness value of 1, the second worst receives a value of 2, etc. and the fittest individual receives a value of J, where $J = M$ is the number of individuals in the population. The probability p_i of an individual to be selected is equal to the new fitness of the individual divided by the total new fitness of all the individuals, as follows:

$$p_i = \frac{f'_i}{\sum_{j=1}^{J} f'_j}. \tag{15}$$

3.4 Concluding remarks

This chapter described the development of the Flock-CC protocol on the basis of the 4 rules that govern the flocking behavior. The performance of the proposed protocol is assessed in Chapter 5 using a number of scenarios with different topologies, traffic loads and instantaneous network conditions (number of idle/source nodes, position of source nodes, failing nodes).

Chapter 4

The Lotka-Volterra-based congestion control (LVCC) approach

4.1 The concept: wireless sensor networks as ecosystems

A WSN is considered to be analogous to an ecosystem. An ecosystem comprises of multiple species that live together and interact with each other as well as the non-living parts of their surroundings (i.e. resources) to meet their needs for survival and coexist. Similarly, a wireless sensor network involves a number of cooperative nodes. Each node has a buffer in order to store packets, a communication channel of a certain (dynamic) capacity, and is able to initiate a traffic flow.

Traffic flows can be seen as species that compete with each other for available network resources (buffer space communication channel capacity) while traversing a set of intermediate nodes forming a multi-hop path leading to the sink. The population size of each species corresponds to the rate of each traffic flow. In analogy with ecosystems, *the goal is the coexistence of traffic flows.*

The LVCC approach as well as the Flock-CC approach provide congestion control and avoidance in a decentralized manner. Decentralization is the process of dispersing decision-making closer to the point of service or action. This is a basic feature of the two proposed approaches that allows flexibility that facilitates self-organization. Such flexibility is facilitated by the lack of dependency on central decision-making. However, it has to be done in a manner that allows some control. This control arises through the interaction between components.

To design a decentralized and autonomic approach, a network is divided into smaller neighborhoods called sub-ecosystems. Each sub-ecosystem involves all nodes that send traffic to a particular one-hop-away node (parent node). *The traffic flows initiated by each node play the role of competing species and the buffer (queue) capacity of the parent node is considered in this thesis as the limiting resource within the sub-ecosystem.*

Within a virtual ecosystem, participant nodes may perform different roles. In particular, each node is able to either initiate a traffic flow i.e. is a source node (SN), or serve as a relay node (RN) to forward packets of multiple flows passing through it, or perform both roles being a source-relay node (SRN). Source nodes are mostly located at the edges of a network (e.g. leaf nodes) while relay nodes are internal nodes (e.g. backbone nodes). The proposed approach provides hop-by-hop rate adaptation by regulating the traffic flow rate at each node. *Each node is in charge of self-regulating and self-adapting the rate of its traffic flow i.e., the rate at which it generates or forwards packets. The traffic flows compete for available*

buffer capacity at their one-hop-away receiving node involved in the path leading to the sink.

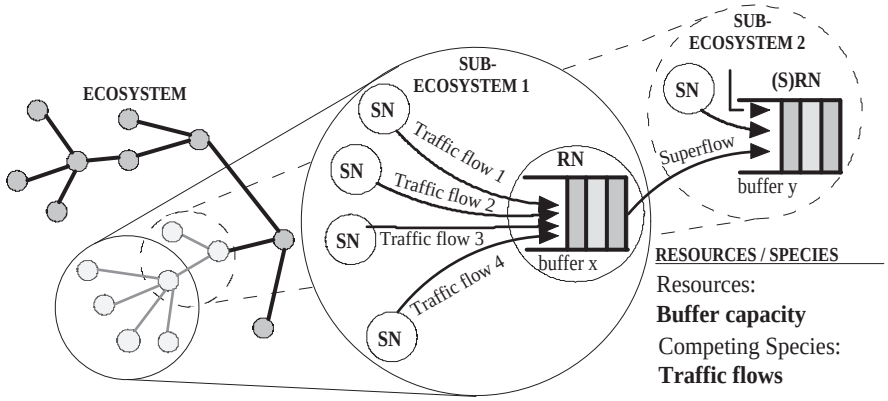


Figure 16: Traffic flows competition in WSNs.

Each sending node is expected to regulate its traffic flow rate in a way that limiting buffer capacities at all receiving nodes along the network path towards the sink are able to accommodate all flows. Thus, the sending rate evolution of each flow will be driven by variations in buffer occupancies of relay nodes along the network path towards the sink. Due to the decentralized nature of the proposed approach, each node will regulate its traffic flow rate using locally available information (i.e. from one-hop away neighbors), thus satisfying the need for low communication overhead.

4.2 Entities of the LVCC approach

This section deals with the roles of the different entities involved in the congestion avoidance mechanism along the path towards a sink: the source node (SN), the relay

node (RN) and the source-relay node (SRN). A SRN acts as both source and relay node, having both functions concurrently operated as described below.

4.2.1 Source nodes (SNs)

Pure **source nodes** are end-entities which are attached to the rest of the network through a downstream node e.g. a relay node (RN), or a source-relay node (SRN) located closer to the sink, as shown in Fig. 17.

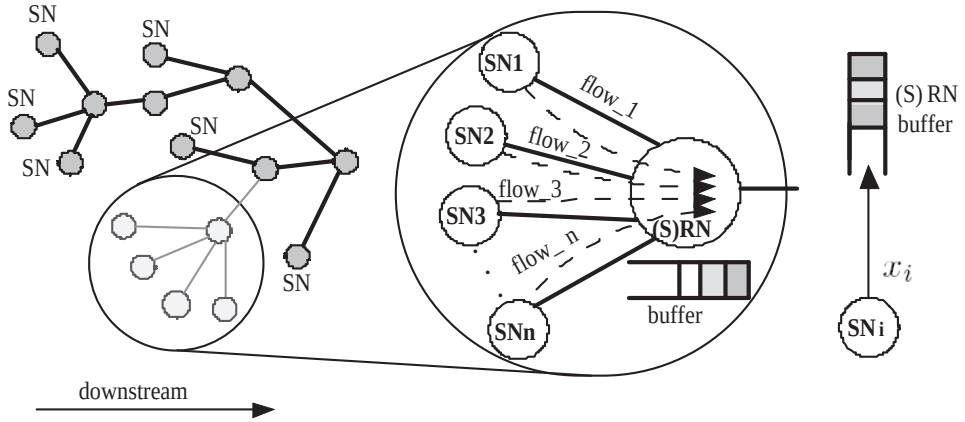


Figure 17: Source nodes competing for a limiting resource at their downstream node.

Each SN is expected to initiate a traffic flow when triggered by a specific event. The transmission rate evolution of each flow is calculated by Eq. 16 (the solution of Eq. 4) that gives the number of bytes sent x_i by flow i. In order to be able to solve Eq. 4 for a single node i, it is necessary to be aware of the aggregated number of bytes sent from all other nodes $\sum_{j=1,j\neq i}^{n} x_j$ which compete for the same resource. This quantity is denoted by C_i. In decentralized architectures, the underlying assumption of C_i-awareness is quite unrealistic. However, each SN can indirectly obtain this

information through a small periodic backpressure signal sent from its downstream SRN/RN (parent node) containing the total number of bytes sent from all parent's children, denoted by BS. Each node can evaluate its neighbors' contribution C_i by subtracting its own contribution x_i from the total contribution BS as expressed by: $C_i = \sum_{j=1,j\neq i}^{n} x_j = BS - x_i$. Thus, Eq. 4 becomes:

$$\frac{dx_i}{dt} = rx_i \left[1 - \frac{\beta}{K}x_i - \frac{\alpha}{K}C_i\right],\ i = 1, ..., n. \tag{16}$$

Eq. 16 is integrated to obtain the calculated transmission rate of each SN, x_i, given by:

$$x_i(t) = \frac{wx_i(0)}{\beta x_i(0) + [w - \beta x_i(0)]\, e^{-\frac{wr}{K}t}},\quad w = K - \alpha C_i \tag{17}$$

The population size of each species i, $x_i(t)$, corresponds to the rate of each traffic flow at time t. In the thesis, the terms flows and species are used interchangeably. As described in Section 2.3.1, the LV model involves four parameters: r, α, β and K. Parameter r corresponds to the speed (how quickly) each traffic flow's rate increases. Parameter α measures the intensity at which the packets of a given traffic flow interact with each other, whereas parameter β measures the intensity at which the packets of different traffic flows interact with each other. Finally, parameter K is the buffer's capacity on each node (the shared resource).

In Section 2.3.4, it was found that the n-species LV system has a global non-negative and asymptotically stable equilibrium point when inter-specific competition is weaker than intra-specific competition i.e. $\beta > \alpha$ and $\alpha, \beta > 0$. This global

coexistence solution is given by Eq. 11, reproduced here for easy reference:

$$x_i^* = \frac{K}{\alpha(n-1)+\beta},\ i = 1, ..., n. \tag{11}$$

A detailed proof of this solution is given in Section 2.3.4.

Furthermore, in order to avoid buffer overflows, it needs to be ensured that when a system of n active nodes converges to the coexistence solution, each node i will be able to send less than or equal to K/n bytes. This is satisfied by Eq. 11 when $\alpha(n-1)+\beta \geq n$ or $\beta - \alpha \geq n \times (1-\alpha)$. If we set $\alpha \geq 1$ and require $\beta > \alpha$ (as imposed by the equilibrium stability condition), then the aforementioned inequality is always satisfied. Therefore, to ensure both convergence and no buffer overflows the following two conditions must be satisfied:

$$\beta > \alpha,\ \ \alpha > 1 \tag{18}$$

The calculated transmission rate of each node, $x_i(t)$, is initiated by $x_i(0)$ and converges to the stable coexistence solution, x_i^* within time T_{conv}. The convergence time, T_{conv}, can be evaluated by $x_i(T_{conv}) = x^*$ (on the basis of Eq. 17) and is found to be proportional to parameter α and inversely proportional to parameters r. This observation practically means that fast convergence can be achieved using small values of α or large values of r, but further discussion is given in performance evaluations.

Each SN evaluates the number of bytes it is able to send using Eq. 17, where a series of values are generated which correspond to number of bytes sent every period T. In Eq. 17, when we set the initial value of the transmission rate $x_i(0)$ and the

current time to 0, we can directly obtain the transmission rate $x_i(t)$ for any time t. Iteratively, if the transmission rate at time kT is $x_i(kT)$ then we can calculate the transmission rate at time $(k+1)T$ by:

$$x_i((k+1)T) = \frac{w(kT)x_i(kT)}{\beta x_i(kT) + [w(kT) - \beta x_i(kT)]\, e^{-\frac{w(kT)r}{K}T}}, \quad w(kT) = K - \alpha C_i(kT) \quad (19)$$

4.2.2 Relay nodes (RNs)

Pure **relay nodes** are entities which do not generate any packets, but forward packets belonging to several flows traversing themselves which compete for their resources.

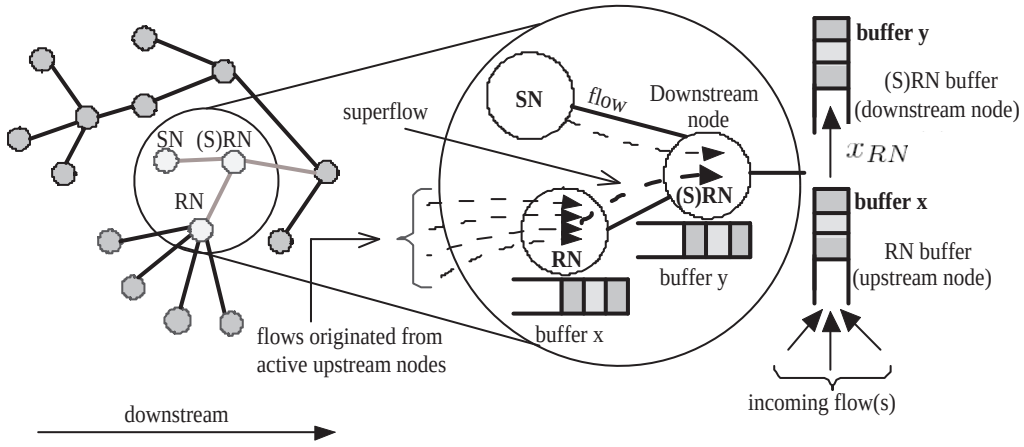


Figure 18: Relay node creates a superflow which competes for downstream node's buffer.

The main function of a RN is to combine (or multiplex) all incoming flows into a superflow and relay it to the dedicated downstream node (SRN or RN) as shown in Fig. 18. Each RN allocates resources for its active upstream nodes based on a

slightly modified expression of Eq. 19 as follows:

$$x_{RN}((k+1)T) = m\left[\frac{w(kT)H(kT)}{\beta H(kT) + [w(kT) - \beta H(kT)]\, e^{-\frac{w(kT)r}{K}T}}\right],$$

where $H(kT) = \frac{x_{RN}(kT)}{m}$, $w(kT) = K - \alpha C^*_{RN}(kT)$ and m is the total number of active upstream nodes which belong to the tree having RN as root. Each RN can calculate the number of its active upstream nodes, m, by examining the source id field of each packet traversing itself. $C^*_{RN}(kT)$ reflects the total number of bytes sent (BS) to the downstream node from all its competing children nodes, subtracting the contribution of a single flow belonging to the superflow, given by:

$$C^*_{RN}(kT) = BS - \frac{x_{RN}(kT)}{m}. \tag{20}$$

4.2.3 Source-relay nodes (SRNs)

A **source-relay node** acts as both source and relay node, having both functions concurrently operated as described above.

4.3 Concluding remarks

This chapter described the development of the LVCC protocol on the basis of the Lotka Voltera population model that governs the coexistence of species competing for limited resources. The performance of the proposed protocol is assessed in Chapter 5 using realistic simulation scenarios of network operation and conditions in order to understand how the variations of the model's parameters influence stability, sensitivity to parameters, scalability and fairness.

Chapter 5

Performance evaluations

This chapter evaluates the performance of the two proposed nature inspired approaches. Each approach is evaluated separatively on the basis of parameter tuning with respect to known networking performance metrics, intrinsic properties and comparison among other related approaches.

5.1 The Flock-based congestion control approach

This section evaluates the performance of the Flock-CC protocol using a number of scenarios with different topologies, traffic loads and instantaneous network conditions (number of idle/source nodes, position of source nodes, failing nodes).

Performance evaluations initially focus on tuning the parameters T and ξ, based on well known networking performance metrics. It is worth noting that by design the behavior of the tunable parameters is well understood, and these were kept at a minimum. Then, the effectiveness of the Flock-CC protocol in mimicking

the bird flocking behavior and its intrinsic properties, in combating congestion was demonstrated. Finally, the superiority of the protocol against related CC protocols was shown. The Flock-CC protocol was evaluated on the basis of the performance measures presented next:

1. *Parameter selection*: The first step was to experiment with the Flock-CC parameters: the values of the spreading variable ξ and the sampling period variable T were selected, based on networking performance metrics as packet delivery ratio (PDR), end-to-end delay (EED) and energy tax. More details on these metrics are given below.

2. *Demonstration of emerging behavior, self-adaptation, robustness against failing nodes and scalability*: The second step highlights the aforementioned properties of the Flock-CC protocol using both visual representations and the performance metrics.

3. *Comparative evaluation*: Finally, the Flock-based protocol was compared against related (nature-inspired and conventional) congestion control approaches.

Results were collected from simulation studies conducted using the ns-2 network simulator [14]. The relevant simulation scripts can be found in [38].

5.1.1 Evaluation setup

5.1.1.1 Topologies

The evaluation topologies consist of a considerable number of nodes which are deployed either in a 2D lattice, or in a random manner, over an area of 400×800 m^2. A 300-node lattice topology is shown in Figs. 19(a) and (b). Lattice topologies of 200 and 400 were also considered. In lattice topologies, neighboring nodes were placed at the same distance ($25 \cdot \sqrt{2}$m) from each other, in such a way that each node has at most 8 neighboring nodes one hop away. In this way, a dense topology was developed with uniform node placement. The lattice topology was used in the overwhelming majority of scenarios so as to better understand and interpret the behavior of the flock-based mechanism. In order to experiment with more realistic topologies, nodes were also deployed in a random manner, as described later.

All nodes were homogeneous and identical in hardware and software setup. The radio propagation range of each node was 50m. A two-ray reflection propagation model was used which accounts for the effect of multipath and fading. The radio model assumes that the radio can lock onto a sufficiently strong signal in the presence of interfering signals. The control packet size and the data packet size were set to 10 and 50 bytes respectively. The CSMA-based IEEE 802.11 MAC protocol, provided in the ns-2 simulator, with an exponential backoff policy was adopted, having RTS/CTS packet exchanges disabled for energy saving and low signalling overhead purposes. Two transmission rates of 2 Mbps and 250 Kbps were used[1]

[1]The 802.11 protocol is not considered by many researchers as not necessarily the best MAC protocol for WSNs (even though it performs well in general wireless ad hoc networks) because of the

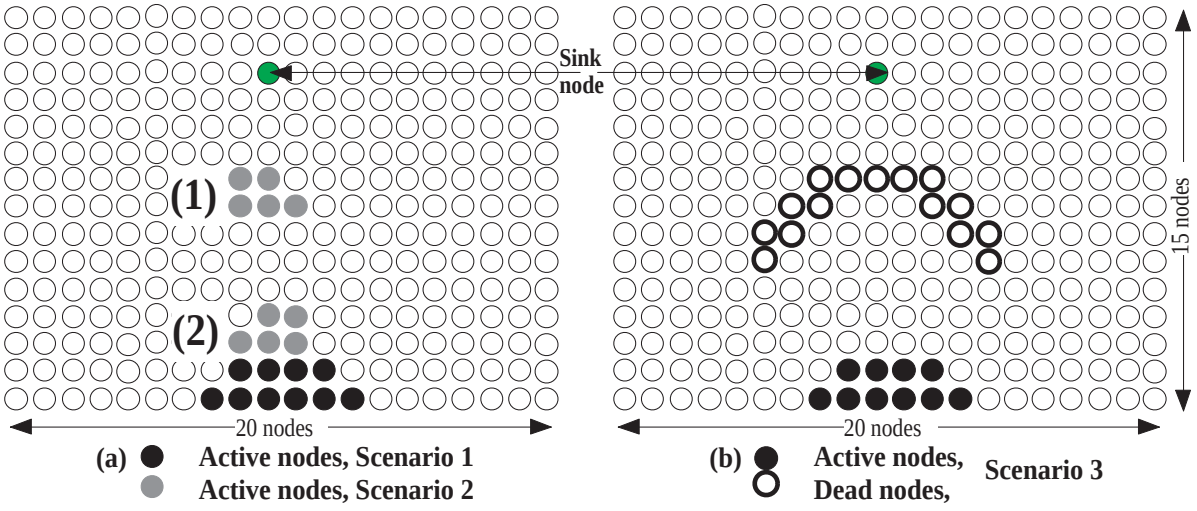


Figure 19: A 300-node evaluation topology with 3 scenarios of operation: (a) Scenarios 1 and 2 with 10 active nodes but different source node placement, (b) Scenario 3 with 10 active nodes and node failures (after 40s). Short description of each scenario is given below.

. In the former case, the queue size of each node was set to 50 packets, while in the latter case it was set to 15 packets. During the communication phase, energy is consumed for both transmitting and receiving data and control packets as well as for idle phases. The power consumption during the idle phase was $712\mu W$, for receiving $35.28mW$, and for transmitting $31.32mW$.

high power consumption and excessively high data rate. However, the IEEE 802.11 MAC protocol was also adopted by many other early research efforts on WSNs as well as on some commercial sensor nodes, like Crossbow's Stargate platform. The apparent unsuitability of IEEE 802.11 MAC protocol for WSNs has motivated several research efforts to design more energy efficient MAC protocols. The most popular commercial MAC protocol is the IEEE 802.15.4, which was specifically designed for short range and low data rate wireless personal area networks (WPAN). Its applicability to WSNs was soon supported by several commercial sensor node products, including MicaZ and Telos. The majority of the MAC protocols proposed for WSNs (a) rely on the CSMA medium access scheme and (b) provide transmission of packets at low data rates in order to save energy [40]. Both features were taken into account in performance evaluations in this thesis. Firstly, the IEEE 802.11 is a CSMA-based MAC protocol, and secondly, both high and low data rates were used in the simulation scenarios. It is worth pointing out that the Flock-CC approach does not depend on the underlying MAC protocol, but uses information available at MAC protocols like queue sizes, the number of successful transmissions and number of retransmissions, if provided. In this thesis, Flock-CC was evaluated over a CSMA-based protocol, whereas its applicability on other medium access schemes as, for example, TDMA is left for future work.

5.1.1.2 Scenarios

The Flock-CC protocol was evaluated under four distinct congestion scenarios to better understand the protocol's behavior and dynamics in responding to the different congestion conditions (persistent and transient hotspots) that could be found in sensor networks. Each scenario corresponded to events occurring in different locations, involving different sets of source nodes and different times of operation. The first three scenarios were related to the 2D lattice topology, while the fourth scenario involved random topologies. All scenarios were tested on topologies of 300 nodes (scenarios $1-3$ are illustrated in Fig. 19), while scenario 1 was additionally tested on topologies of 200 and 400 nodes. All simulation scenarios were run for a period of 100 seconds.

The *first scenario* (scenario 1) emulated persistent congestion phenomena and involved a set of 10 closely placed source nodes shown as black shaded in the middle bottom part of Fig. 19(a). The same setting was used in the larger topologies. In many real-world WSNs applications, it is quite common to have source nodes which are closely placed to each other being activated almost at the same time when an external event (e.g. a disaster-related event such as fire, earthquake) is detected. In these cases, persistent hotspots are created in the vicinity of source nodes.

The *second scenario* (scenario 2) emulated transient congestion phenomena. Initially, the front set (1) of 5 grey shaded source nodes of Fig. 19(a) (close to the sink) were sending packets. In this way, a small, transient congestion hotspot (of heavily loaded queues and wireless channel) was created around and in front of

source nodes towards the sink. In order to demonstrate the ability of the packets to manoeuvre around the hotspot, the bottom set (2) of 5 grey shaded source nodes started sending packets at $t = 50$s. At $t = 70$s, the upper set of nodes (1) stopped sending packets.

The *third scenario* (scenario 3) emulated an extreme scenario of node failures. The black shaded source nodes located in the bottom part of the network of Fig. 19 (b) were initially sending packets (as in the first scenario), while bold circled nodes in the middle of the network failed at $t = 40$s. The aim was to demonstrate the ability of the packets to manoeuvre around a dead zone. The shape of the dead zone (involving failed nodes) was chosen so as to demonstrate the ability of the algorithm to manoeuvre around areas where packets can get trapped and as such cannot move forward.

Finally, the *fourth scenario* (scenario 4) was used to demonstrate the ability of the Flock-CC protocol to also perform well when the underlying sensor network infrastructure is random. This scenario involved two randomly generated uniform topologies with 10 closely placed source nodes and 1 sink node. The sink was randomly selected from nodes in the network. The two random topologies were: (a) a sparse topology of 175 nodes where the average node degree[2] was 4.72 and (b) a dense topology of 300 nodes where the average node degree was 8.85. An indicative figure of the random topology used is shown below, and further details can be found in [38].

[2]The node degree is the number of connections the node has to other nodes in the network.

All scenarios were used for parameter selection and comparative evaluations among all approaches. Additionally, the second scenario was used to evaluate the emerging behavior and the self-adaptive behavior of the Flock-CC model. Furthermore, the third scenario was chosen to demonstrate the robustness of the Flock-CC protocol against failing nodes, even under extremely undesirable situations.

5.1.1.3 Variables

The spreading variable ξ was chosen to range from 0 to 1 to allow experimenting with zero spreading to wide spreading of the flock. The time between successive control packets, T (sampling period), was assigned the values $0.5, 1.0, 1.5$ and 2.0s in all approaches. The selection of T to be less than or equal to 2s was guided by the desire to maintain responsiveness to changes in the network state. It was also desirable to avoid overwhelming the network with control packets, thus T should not be very low, hence $T \geq 0.5$ was also selected. T_{lost} was set at $3T$.

In all scenarios, three different traffic loads were considered: low, high and extreme. At high MAC transmission rates (2 Mbps), each sender node was allowed to generate constant bit rate (CBR) traffic of either 25 (low load), or 35 (high load), or 45 (extreme load) pkts/s when triggered by an event. These three cases were considered as slightly congested, congested, and heavily congested network conditions, respectively. In each scenario, all nodes were sending at the same rate. The corresponding traffic rates for the transmission rate of 250 Kbps was $10, 15$ and 20 pkts/s.

5.1.1.4 Related approaches for comparison

From the perspective of network layers, the Flock-CC approach provides multi-path routing and congestion control capabilities for WSNs. Flock-CC was compared to five protocols that can be adopted during a congestion crisis period: (a) a conventional congestion-free multi-path routing protocol based on shortest paths (baseline scenario), (b) a typical congestion-aware routing protocol that routes packets over multiple paths, choosing each time the least congested next hop node in terms of queue length (among nodes involved in the shortest paths to the sink), (c) a well known single-path routing protocol for adhoc networks, AODV [78], (d) an ant-based algorithm for multi-path reactive and proactive routing in mobile ad-hoc networks which is based on ideas from Ant Colony Optimization, AntHocNet [12], and (e) an ant-based multi-QoS routing protocol, AntSensNet [13]. These strategies/protocols are summarized in Table 1.

Table 1: Congestion control strategies for comparative analysis.

Strategy	*Congestion detection*	*Routing scheme*
Flock-based Congestion Control (Flock-CC)	Queue loading, MAC layer collisions and retransmissions	Multi-path
No Congestion Control (NCC) shortest paths	None	Single-path
Congestion-Aware Routing (CAwR)	Queue occupancy	Multi-path
AODV [78]	None	Single-path
AntHocNet [12]	MAC layer activity	Multi-path
AntSensNet [13]	MAC layer activity	Multi-path

5.1.1.5 Metrics of performance

Three performance metrics for congestion control approaches were taken into account: the packet delivery ratio (PDR), the mean end-to-end delay (EED), and the energy tax. Each performance metric is further discussed below:

- **Packet delivery ratio (PDR)** is defined as the ratio of the total number of data packets received at the sink (control packets are disregarded) divided by the total number of data packets generated by all source nodes (destined to the sink). The following formula used to calculate PDR at the sink at the end of each simulation scenario:

$$PDR = \frac{P_r}{P_g}, \tag{21}$$

  where P_r is the number of data packets received at the sink (from all source nodes) and P_g is the number of data packets generated by all source nodes throughout the scenario duration.

- **Mean End-to-end delay (EED)** is defined as the mean time taken for a data packet to be transferred across the sensor network from the source node to the sink. If, at any part of the network, a data packet is retransmitted by the MAC protocol after facing collision, the additional delay is included in the EED. The end-to-end delay is calculated over all surviving data packets that reach the sink throughout a simulation scenario. Packets that are lost due to either buffer drops or wireless channel collisions (after a finite number of retransmissions) can not be considered. The following formula used to

calculate EED at the sink at the end of each simulation scenario:

$$EED = \frac{\sum_{i=1}^{P_r}(T_r^i - T_s^i)}{P_r}, \tag{22}$$

where T_r^i is the time that each packet i was received at the sink and T_s^i is the time that each packet i was generated at the source node. In scenarios with high packet losses, where only a small number of packets are delivered at the sink, the EED metric may not be accurate enough.

- **The energy tax** metric is defined as the mean energy consumption per node per delivered data packet, measured in mJoules/delivered packet at the sink. The energy tax is calculated by the following formula:

$$\text{Energy Tax} = \frac{\sum_{i=1}^{N} E_c^i}{N \times P_r}, \tag{23}$$

where N is the number of nodes within the sensor network, and E_c is the energy consumed at each node i throughout the simulation scenario.

5.1.2 Parameter selection

This section evaluates the performance of the Flock-CC protocol and provides parameter selection guidelines. Only topologies of 300 nodes are discussed in this section. Results regarding larger topologies are presented later. Each scenario, using different combinations of parameter values, was executed 30 times and the mean values of the metrics over all scenarios are presented below. In selected figures, the mean values are supplemented with 95% confidence intervals. When dealing with non-normal distributions, 'confidence intervals' are used as a measure of variability

instead of using the standard deviation. A 95% confidence interval contains the middle 95% of the numbers in a list.

The Flock-CC protocol development was investigated in several stages (detailed results and conclusions can be found in [38]). The notions of the FoV, the magnetic field and randomness were progressively added to the protocol. Each element contributed a significant improvement in the performance of the Flock-CC protocol. In the absence of orientation and attraction to the sink (no FoV and magnetic field), packets were wandering around in the network and got trapped in routing loops. Eventually, the overwhelming majority of packets were lost due to collisions in the wireless channel. The introduction of the FoV and the magnetic field caused a steep rise in PDR of approximately 55% (in scenario 1). Increasingly, the introduction of perturbation provoked further increase in PDR of up to 10% (scenario 1).

More specifically, perturbation was realized on the basis of two random selection techniques: roulette wheel selection and rank based selection. Rank based selection was shown to outperform roulette wheel selection, exhibiting slightly higher PDR values $(1 - 3\%)$ across all three scenarios, for almost every value of T and ξ. Thus, the rank based selection was incorporated in the Flock-CC protocol.

Below, we present the results and discuss the performance of the final model of the Flock-CC protocol involving all flocking characteristics.

5.1.2.1 Packet delivery ratio (PDR):

Fig. 20 illustrates the PDR for all three scenarios of Fig. 19 with respect to parameters T and ξ. The results presented here consider sending nodes generating traffic at the rate of 35 pkts/s (high traffic load, i.e. the network can be considered congested). Similar observations apply for low (25pkts/s) and extreme (45 pkts/s) traffic loads, thus the figures were omitted.

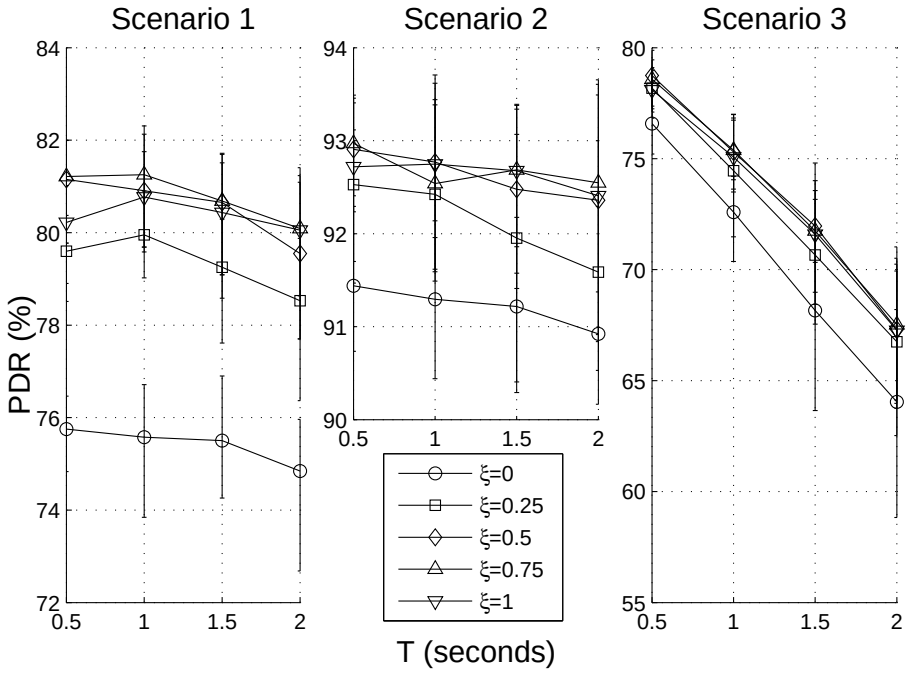


Figure 20: Packet delivery ratio (PDR), Flock-CC with all flocking characteristics, all scenarios (high traffic load). The vertical lines indicate 95% confidence intervals.

In scenarios $1, 2$ and 3, the highest value of PDR (around 82%, 93% and 78% respectively) was observed for $\xi = \{0.5, 0.75\}$. Significant performance deterioration was exhibited for low spreading values, $\xi = \{0, 0.25\}$. In the first scenario, poorer performance was also observed for high spreading, $\xi = 1$, but not at the same scale.

At low ξ values, weak attraction was exercised by each idle node, even if these nodes were able to accommodate incoming traffic load. Since idle nodes are usually found at the borders of the packet flock, low ξ values 'force' packet flocks to move in a coherent formation. In this case, a number of available paths were left unexploited while other (popular and perhaps, shorter paths to the sink) faced overloading, thus resulting in low PDR. On the other hand, at very high ξ values (close to 1), each idle node was exercising strong attraction (causing high packet spreading towards the border of the flock). In this case, two problems may occur: (a) There is no evidence of the channel quality in the vicinity of each idle node, thus if the channel is busy a high number of losses may arise. (b) A large number of packets are attracted to the borders of the flock (left and right), where the majority of idle nodes reside. At the same time, these packets are also attracted to the sink due to the inherent attraction of the magnetic field. Thus, packets at the borders of the flock follow a diagonal path (towards the sink) and collide with ongoing packets 'flying' through the center of the flock towards the sink, resulting in a lower PDR. The slight problem with high ξ values was more pronounced in the first scenario, where packet spreading was apparent on a larger scale.

An interesting observation is that, for scenarios 1 and 2, PDR reached around 82% for $T = 1$s, while it exhibited slight decrease for $T = 1.5$s and $T = 2$s. On the other hand, for the third scenario (failing nodes), the PDR ranged from 76% to 79% for $T = 0.5$s for ξ values of 0 to 1, with steep decreasing trends of $10 - 11\%$ as T approached 2s. For low T values where control packets were broadcasted quite

frequent, the Flock-CC mechanism was kept updated regarding the network state. Frequent updates are of prime importance in scenarios with failing nodes. This of course happens at the cost of higher energy expenditure-overhead, however with a reduced number of retransmissions expected.

In scenario 2, lighter congestion phenomena occurred compared to scenarios 1 and 3 due to the low number of closely located sending nodes. Traffic load injected into the network in the two small hotspots of scenario 2 did not overload the network resources as much as the injected traffic load in the hotspot of scenarios 1 and 3.

5.1.2.2 Packet loss:

In scenario 1 (Fig. 21), the number of collisions dominated the number of buffer overflows for all values of T and ξ, while the overwhelming majority of packets were lost within the hotspot area.

Quite often, due to the shared wireless medium, packets collide before reaching the receiver node. Thus, buffers are less often filling up. The coherent packet movement (through popular nodes in the shortest paths) observed at low ξ values (close to 0) resulted in quite high number of collisions. This is because packets were traversing a small, and closely located (correlated), set of paths toward the sink causing overloading of the limited wireless channel resources. In CSMA-based MAC protocols, a detection of a collision causes retransmission of the collided packet. Therefore, the increase of collisions causes in increase in the number of retransmissions.

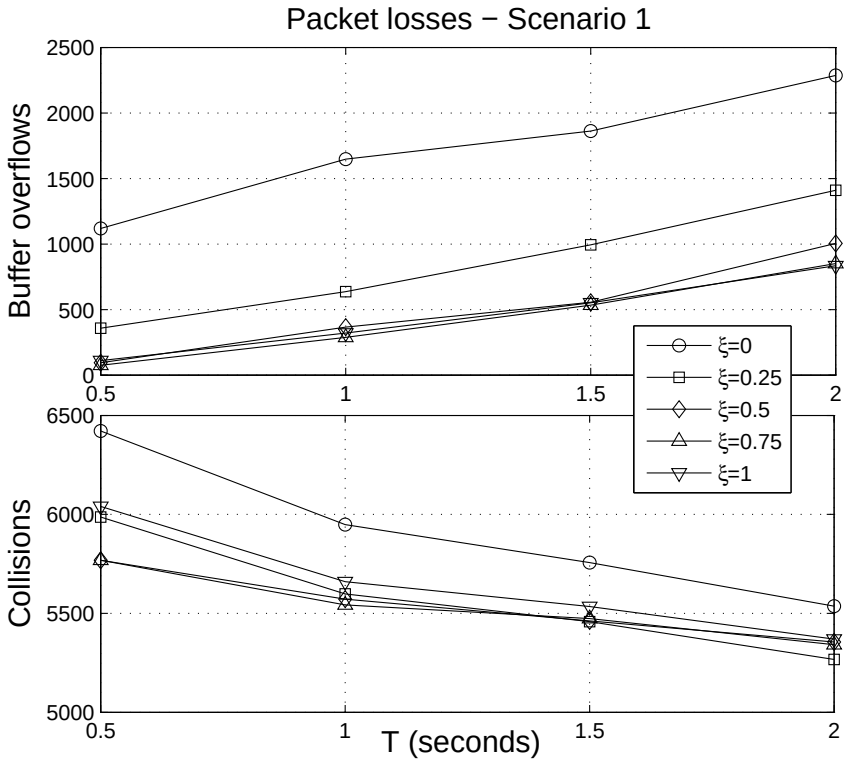


Figure 21: Buffer overflows and collisions, scenario 1, high load.

Fig. 22 illustrates the ratio of retransmissions over successful transmissions in scenario 1 ($T = 0.5$s, 35 pkts/sec) and reveals the high number of retransmissions at $\xi = 0$ and 0.25. An important observation is that the number of retransmissions exceeded the number of successful transmissions for every value of ξ. This is attributed to the high traffic load injected into the network causing overloading of wireless channel. At medium ξ values $(0.5, 0.75)$, packets spread over uncorrelated paths minimizing the number of collisions, and thus retransmissions. The effect of high packet spreading at $\xi = 1$ resulted in a slight increase of collisions. This is due to the fact that packets were allowed to move (parallel to the sink) to nodes at equal hop distance from the sink, being able to get trapped in loops.

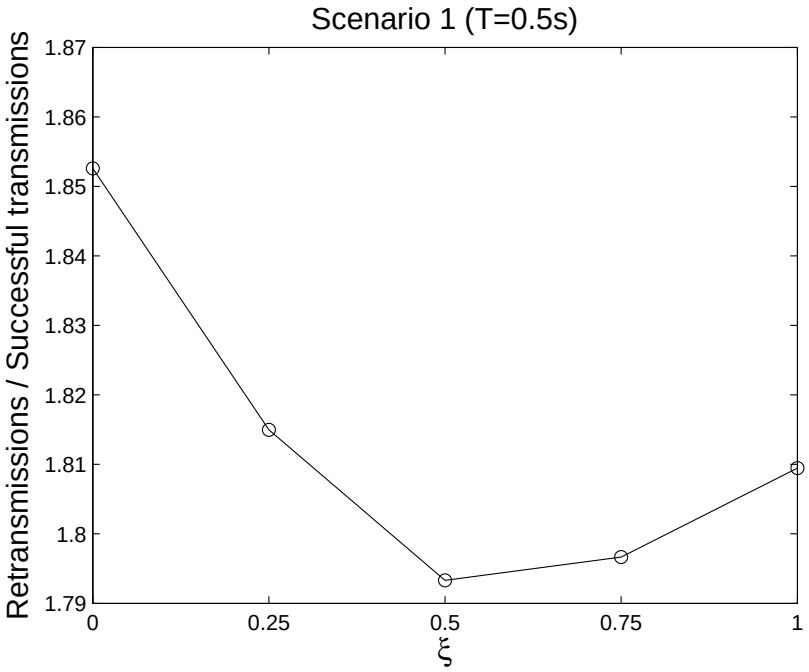


Figure 22: Ratio of retransmissions over successful transmission, scenario 1, $T = 0.5$s, high load.

For scenario 1, the lowest *total* packet loss was observed at $\xi = \{0.5, 0.75\}$. As shown in Fig. 20, these two ξ values consistently achieved the best performance in terms of PDR. The reason for the efficiency of these two ξ values is found in the balanced way of packet movement between coherent formation and spreading. The attraction to idle nodes is strong enough to exploit their resources (buffer space, channel capacity) without causing overconsumption.

Another observation in scenario 1 is that, the increase of T provoked an increase in the number of buffer overflows and a reduction in the number of collisions and retransmissions. The reason is the following: recall that the desirability function is evaluated once every sampling period T and the calculated desirability values of

neighboring nodes are used throughout the sampling period. At high T values, the desirability values are evaluated infrequently. Packets sent within the same sampling period choose, with high probability, the same set of hosting nodes (those having high desirability values). Since the sampling period is longer, the set of desirable (and most preferable) nodes (within each sampling period) becomes overloaded, facing high queue occupancy. In addition, the preference of packets to the same new hosting nodes eliminates packet spreading and reduces collisions. On the other hand, at low T values, packets spread more frequently to different new hosting nodes eliminating buffer overloading but causing an increase of channel collisions around the chosen nodes.

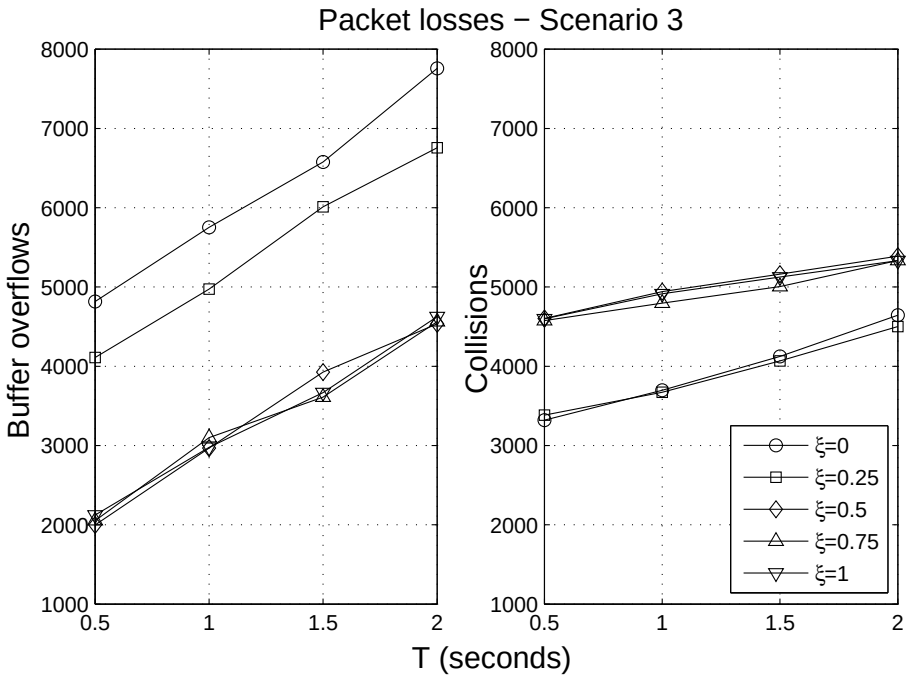


Figure 23: Buffer overflows and collisions, scenario 3, high load (major node failures).

Fig. 23 shows buffer overflows and collisions in scenario 3 where major node failures were simulated. In this scenario, it is worth noting that the majority of packets were lost around failing nodes. As can be seen, buffer overflows followed the same increasing trends with the increase of T as in scenario 1. On the other hand, in scenario 3, as the value of T increases the number of collisions increased, as opposed to scenario 1 where the number of collisions decreased. Also, it can be seen that there was a steep rise in the number of collisions (as well as in the number of buffer overflows) as T approached 2s. The reason is that the slow-paced mechanisms of control packets exchange and desirability evaluation made the flock of data packets incapable of adapting to the rapidly changing network conditions. For example, the high number of packets destined to the sink through the dead zone is expected to cause fast exhaustion of wireless channel (and buffer) capacity around the nodes one or two hops away from 'dead' zone area. This was the case at high T values. On the other hand, at lower T values, fast-paced exchanges of control packets and evaluation of desirabilities 'forced' the flock of packets to quickly manoeuvre around the zone of 'dead' nodes and let to a significant reduction of packet losses. Based on Fig. 23, choosing $T = 0.5$s is definitely better for this metric in the case of failing nodes.

Overall, scenario 3 exhibited higher number of buffer overflows and lower number of collisions than scenario 1 for almost all values of T and ξ. This is because in scenario 1 packets were spread in the network throughout the duration of the scenario causing higher collision rates. On the other hand, in scenario 3, the incidence of

node failures led data packets to find alternative paths to the sink by bypassing the 'dead' zone as shown in Fig. 31. In this case, the movement of packets through the new paths to the sink was necessarily very cohesive due to the new topology, which resulted in this particular example. Thus, packets reaching a diagonal position from the sink had only one (highly attractive) node in their FoV placed closer to the sink. The attraction of packets to nodes along the diagonal led to a very narrow packet movement and thus, high number of buffer overflows and a lower number of collisions.

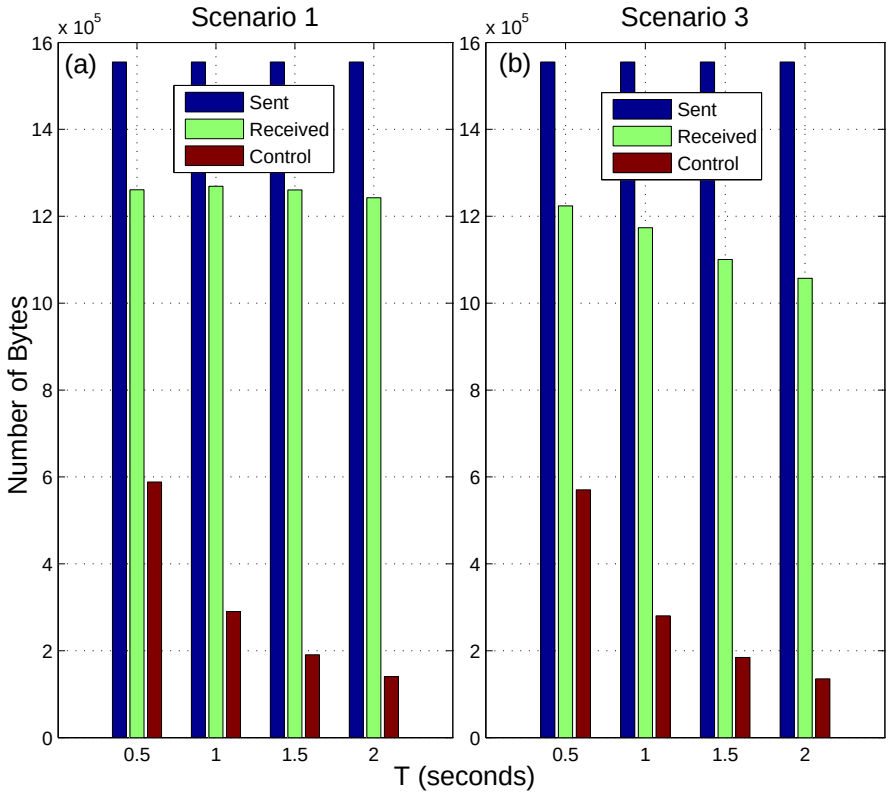


Figure 24: Total number of bytes in data packets sent, data packets received and control packets sent, in scenarios 1 and 3 for different values of T when $\xi = 0.75$, high load.

Figs. 24(a) and (b) illustrate the total number of bytes in control packets sent from all 300 nodes in the network throughout the simulation scenario (around 100 sec) over the total number of bytes sent (injected into the network from source nodes) and received (at the sink) for all values for all values of T (when $\xi = 0.75$) in scenarios 1 and 3 respectively, for 35 pkts/sec. Recall that in our scenarios the size of each control packet was 10 bytes and the size of each data packet was 50 bytes.

It is worth pointing out that the total number of control packet overhead depends solely on the network configuration (data and control packet sizes, number of nodes, value of T, simulation duration) and it is independent of the number of active nodes and of the amount of data packets injected into the network. For example, the number of control packets transmitted on a network of 300 nodes when $T = 0.5$s and the simulation duration is 100s is $60 * 10^3$ packets. This amount of control packets is needed for network (route) maintenance.

Fig. 24(a) shows that, in scenario 1, the control packet overhead exhibited a steep decline as T approached to 2. An important observation is that, at the same time, the number of bytes received at the sink exhibited only a very slight decrease. Therefore, there was no need for fast updates (and thus frequent control packet transmissions) as previously discussed for scenario 1. The control packet overhead is expected to be significant at lower traffic loads and to to become negligible at higher traffic loads.

Fig. 24(b) confirmed the fact that frequent control packet transmissions are needed in scenarios involving failing nodes. More specifically, the number of bytes

received at the sink in scenario 3 exhibited significant reduction (resulting in lower PDR) as the number of control packets (and bytes) decreased (with the increase of T). Therefore, even though the packet control overhead is high at low T values, frequent control packet transmissions are necessary in order to achieve high PDR. This observation emphasizes the need for adapting T in accordance to the network (failure) conditions, which is discussed below. In failure-free scenarios, higher T values can be considered in order to avoid high control packet overhead.

5.1.2.3 End-to-end delay (EED):

As shown in Fig. 25, from the viewpoint of EED, comparably low delays were observed for $\xi = 0.75$, especially in scenarios 1 and 3. In scenario 1, the Flock-CC protocol achieved the lowest EED ($\approx$ 335ms) for $\xi = 0.75$ and $T = 0.5$s. In scenario 3, the selection of T from 0.5s to 2s was fairly insensitive with marginal gains for optimally tuning T.

As can be seen in all scenarios, but with emphasis in scenarios 1 and 3, the values of EED for $\xi = \{0, 0.25\}$ were significantly high compared to other ξ values. High EED values were observed for both low and high T values. At low T values, frequent evaluations of desirabilities were performed and thus desirable next hop nodes were changing at a fast pace. At the same time, due to low ξ values, packets were 'forced' to move in coherent packet flock formations, i.e. choosing next hop nodes belonging to a very small number of closely located paths to the sink. The proximity of these paths led to very high number of collisions, and thus retransmissions. At high

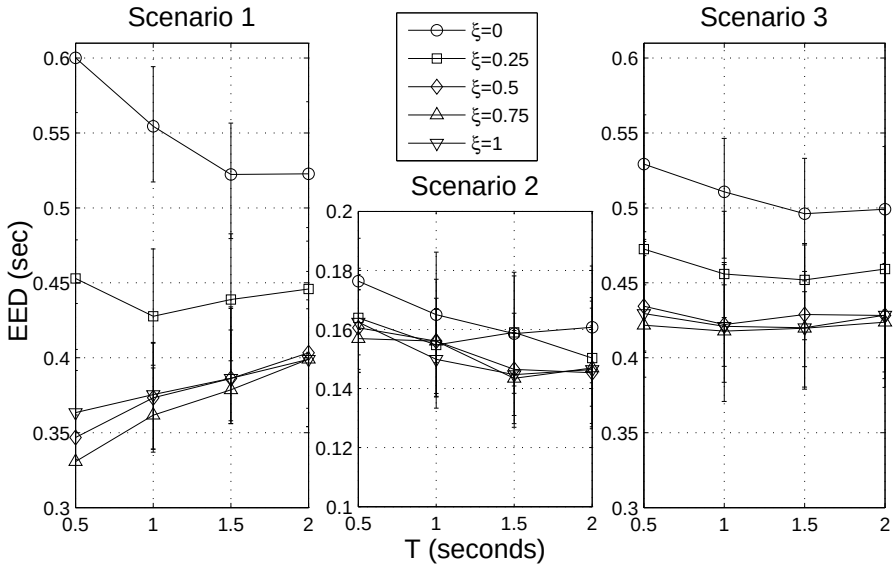


Figure 25: End-to-end delay (EED), Flock-CC with all flocking characteristics, all scenarios (high traffic load). The vertical lines indicate 95% confidence intervals.

T values, desirable next hop nodes were changing at a slow pace. The coherent movement of packets led to fast consumption of buffering capacities on the small set of desirable nodes, and progressively to high number of buffer overflows (see Fig. 23), and thus retransmissions. As a result, increased queueing delays and were observed. Results for scenario 1 show that at higher ξ values ($\xi = 0.5, 0.75, 1$), faster transitions of packets to the sink were observed for low values of T ($T = 0.5$s). The reason is that frequent changes in desirable nodes (low T) are effective when packet spreading is enabled (medium to high ξ values) and a high number of paths to the sinks are available. Under these conditions, individuals in the flock are allowed to exploit the whole space and move on a balanced way over multiple paths to the sink. And this, in turn, leads to reduced number of buffer overflows and collisions, and thus fewer retransmissions and lower EED.

5.1.2.4 Energy tax:

Fig. 26 shows the results concerning the energy tax spent per delivered packet. The first observation is that the least energy tax was paid in the second scenario, where packets traveled shorter paths to the sinks on average since the half of active nodes were located close to the sink. The highest energy tax was paid in scenario 3 (with failing nodes) due to the fact that packets traveled longer paths to the sink on average in an effort to manoeuvre around the 'dead' zone of failing nodes.

Results confirmed the intuitive reasoning (made earlier) that frequent updates at low T values led to higher energy tax in scenarios with no failing nodes (scenarios 1 and 2). The highest energy tax per delivered packet in the aforementioned scenarios was spent for $T = 0.5$s while less energy was consumed for $T = 1.0 - 2.0$s. However, the difference was marginal reaching up to only $1.5\mu J$ per delivered packet. The changes in energy tax values were fairly insensitive to ξ (besides for $\xi = 0$), but the values $\xi = \{0.25, 0.5, 0.75\}$ exhibited better behavior.

In scenario 3, low T values led to less energy expenditure. This is because frequent updates at low T values was the key to the fast establishment of alternative paths around the 'dead' zone and to the reduction of wandering around packets.

Taking all the results of this section into consideration, a good compromise value for ξ is 0.75. The value of T can be set to $T = 0.5$s since the gain from minimizing buffer overflows (observed at $T = 0.5$s) is larger than the gain from minimizing energy expenditure (observed at $T = 2$s). Nevertheless, if slight energy gains are of utmost importance, parameter T can be set to higher values. Alternatively, an

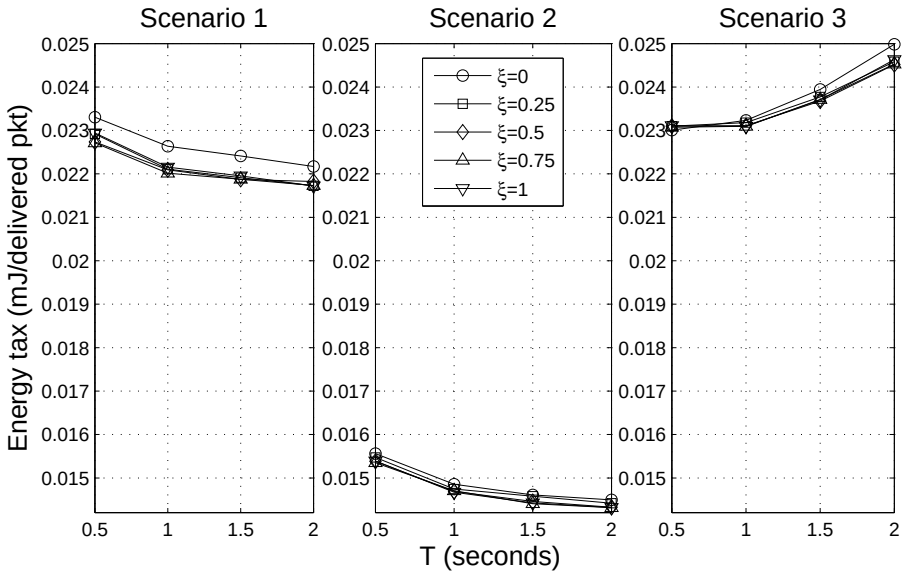


Figure 26: Energy tax spent per delivered packet, Flock-CC with all flocking characteristics, all scenarios (high traffic load).

adaptive mechanism can be used to dynamically adjust the value of T according to changes in the network state. A set of simple guidelines for tuning parameter values is given at the end of this section.

Based on the aforementioned compromise values for ξ and T, the achievable throughput of each active node per scenario measured at the sink is shown in Fig. 27. More specifically, the number of packets per second received at the sink from each one of the 10 active nodes is illustrated. As can be seen, in scenario 1, Flock-CC achieved resource fairness between flows, meaning that all flows (initiated from active nodes) got roughly the same amount of buffer space and wireless channel capacity on each node along the paths (following the packets) toward the sink. In particular, $25 - 30$ packets per second on average were received at the sink from each active node.

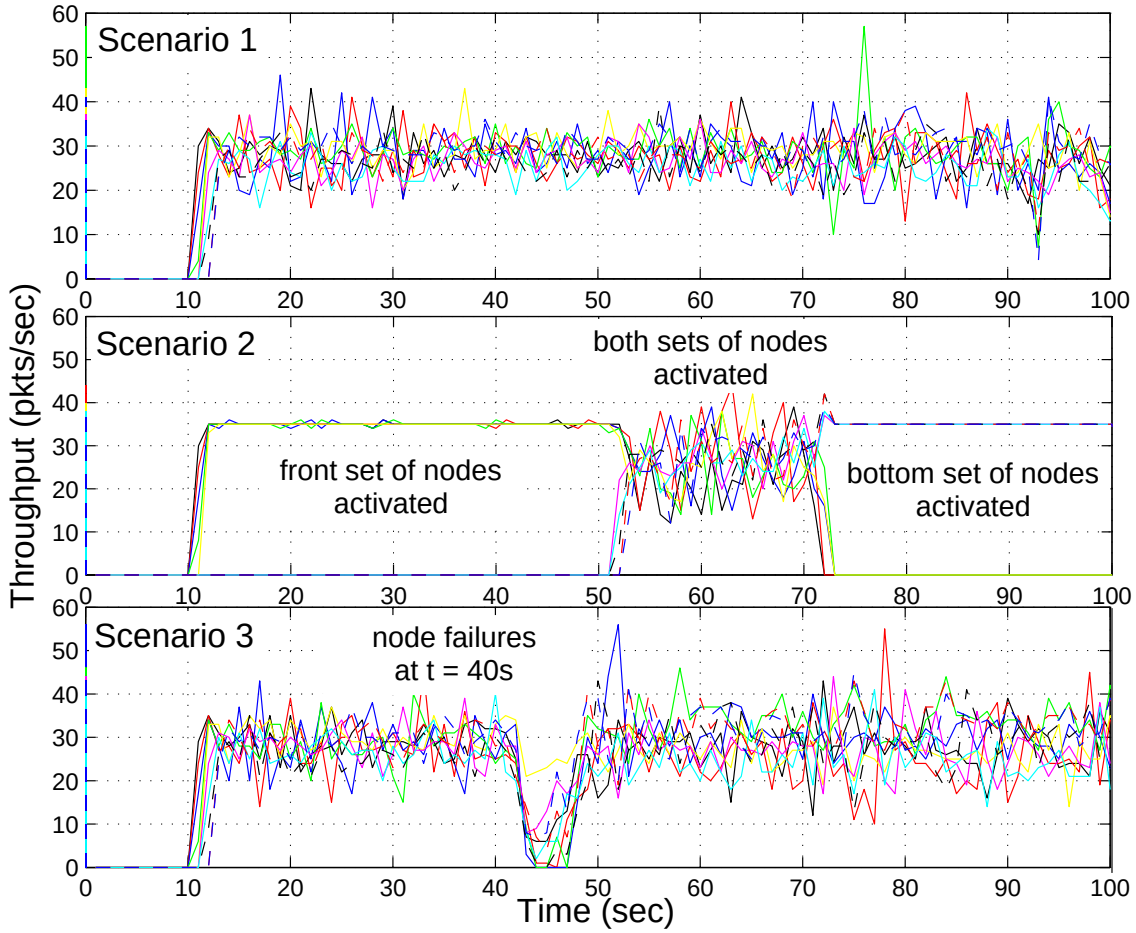


Figure 27: Throughput of each active node measured at the sink when $\xi = 0.75$ and $T = 0.5$s, all scenarios (high traffic load).

In scenario 2, the front set of 5 nodes were initially (from $t = 10$s to $t = 50$s) activated. Almost all 35 pkts/sec were received at the sink during this time period. When the bottom set of 5 nodes was activated (from $t = 50$s to $t = 70$s), around $25 - 30$ pkts/sec were 'fairly' delivered at the sink from each one of the 10 active nodes.

Results for scenario 3 are similar to those of scenario 1, except for the time period between $t = 40$s and $t = 50$s. During this time period, the throughput

of almost all active nodes exhibited a steep decline as a result of node failures at $t = 40$s. However, results show that approximately 10s after the catastrophic event that caused a temporary disruption of 'established' paths to the sink, Flock-CC allowed for throughput recovery of all active nodes to the same levels as before failures.

5.1.2.5 Low data rate WSNs:

The applicability of the Flock-CC protocol on low data rate WSNs (e.g. 250 Kbps) was also evaluated on the basis of scenario 1 to 3. Low, high and extreme traffic loads were considered. The results were similar to those obtained with high data rates of 2 Mbps (see Fig. 20). The most significant difference was that in low data rate WSNs the overwhelming majority of packet losses were attributed to collisions. Due to the low transmission rates, buffers were occasionally filling up, and rarely leading to overflows.

5.1.2.6 Random topologies:

The performance of the Flock-CC protocol was also evaluated on sparsely and densely deployed random topologies (scenario 4). The results for the dense topology in terms of PDR and EED are illustrated in Fig. 28.

The traffic load transmitted from each sender was 25 pkts/s (low load). Due to the high density of nodes (average node degree= 8.85), the PDR was around 10% lower compared to the grid topology of scenario 1 (node degree= 8). As expected, the reduction in PDR was expected due to the extremely high number of collisions.

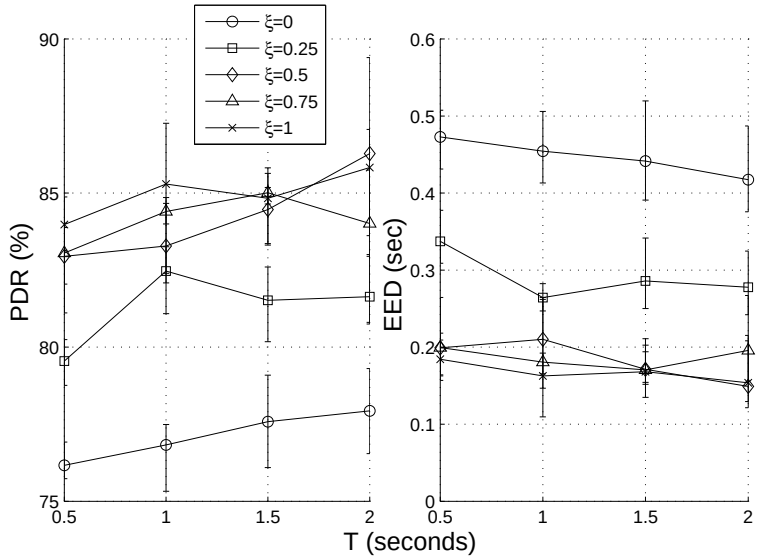


Figure 28: Packet delivery ratio (PDR) and end to end delay (EED), scenario 4 (random dense topology), 25 pkts/s.

The highest PDR was achieved for $\xi = 0.5$ and $T = 2$s, while acceptable levels of PDR were achieved for $\xi = \{0.75, 1\}$ and $T = 1 - 2$s. The same combination of values led to a low EED. The decrease in the number of collisions as T approached 2 was reflected in the increasing trends of PDR and the decreasing trends of EED. There was a concurrent, but slight increase in the number of overflows which did not have a major influence on PDR and EED.

Results for the sparse topology (average node degree= 4.72) are shown in Fig. 29. As can be seen, the PDR was reduced by up to 20% compared to the dense topology while the EED exhibited increases of up to 0.5s. The poor performance experienced in the sparse topology was attributed to the limited network resources that led to the steep increase of buffer overflows. More specifically, in the sparse topology, buffer overflows increased by almost 10 times regardless of the values of

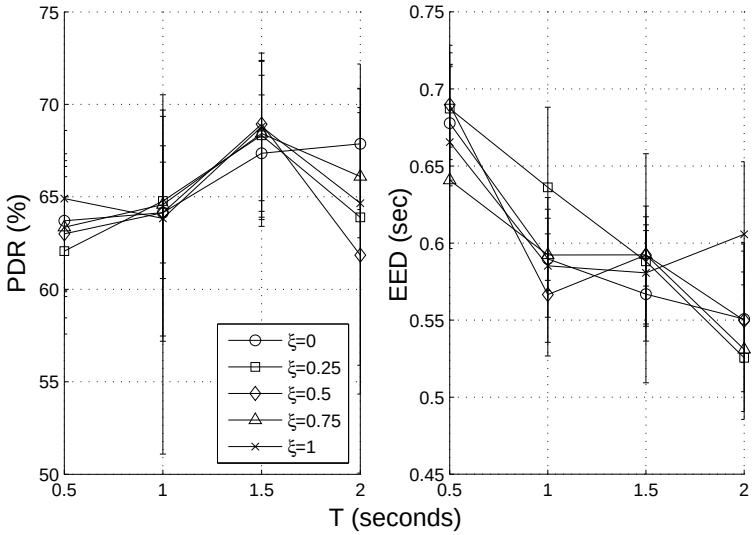


Figure 29: Packet delivery ratio (PDR) and end to end delay (EED), scenario 4 (random sparse topology), 25 pkts/s.

T and ξ. This behavior was expected, since the sparse placement of nodes forced packets to move through a small number of paths to the sink. Therefore, the traffic load injected into these paths could not be accommodated by the limited buffer capacity of nodes in these paths. This is of course a design and dimensioning issue, and for example increasing buffer capacity can improve the performance in terms of PDR at the expense of longer delays (higher EED).

5.1.2.7 Recommended setting of ξ and T:

The results for dense and sparse topologies suggest that a good compromise value for parameter ξ should be either 0.5 or 0.75. The latter value seems more preferable since it was found to be effective and efficient in scenarios with grid topologies with

low and high MAC transmission rates. *For the rest of this study, the value of* ξ *is set to* 0.75.

On the other hand, the value of T can be set according to the network characteristics and conditions. More specifically, results showed that in a controlled environment (e.g. a predefined topology) like a grid topology, for network devices with characteristics as in the simulations, the value of T can be initially set to 1s (not too small to avoid high packet overhead and not too high to maintain responsiveness to changing traffic conditions) while a reduction to 0.5s is required in the presence of failures to allow for fast recovery. Furthermore, in random topologies, a good compromise value for T was found to be 2s for maximizing PDR and minimizing energy consumption at the same time. Therefore, an adaptive mechanisms for tuning T taking all these into consideration is of prime importance.

An initial attempt towards developing an adaptive mechanism for T was implemented and some first results are presented below. In the simple adaptive mechanism, the value of T was initialized to 1s. In the case of a failure, the value of T was reduced to 0.5s for all nodes one hop away from a failing node. After 2s with no other node failures sensed, T was reset to 1s for energy saving purposes. Scenario 3 involving a set of failing nodes was considered when evaluating the adaptive mechanism. Results showed that the achieved PDR was quite similar as in scenarios for $T = 0.5$s. Even though, the simple adaptive mechanism seems quite promising, design choices for optimally tuning T for different kind of traffic rates, network topologies, and network conditions need further study. More specifically,

there is a need for defining which nodes will get involved in decreasing their T values (e.g. number of hops from a failure point), for how long, and whether the variation of T values is to be done on a rule basis or on an equation basis. Therefore, further study of adaptive techniques is left for future work. *For the rest of this study, the value of* T *is set to* $0.5s$.

5.1.3 Emergent behavior of group-level transitions

This section investigates the emergent behavior of the collective motion of packet flocks through the network using the scenarios $1-3$. The emergent behavior can be perceived: (a) directly by the visual representation of flock movements and (b) indirectly on the basis of the performance evaluation metrics. The visual representations of Figs. 30 and 31, using the Flock-CC protocol, depict the sink node in the upper side of each figure, while active nodes are highlighted by bold circles. The number of packets visiting each node within a 1-second time slot snapshot is indicated inside every node. Darker colors indicate higher number of packets.

Evaluation results based on scenario 2 and high traffic load demonstrated the obstacle avoidance behavior of packet flocks. Fig. 30(a) depicts the bird-like motion of packets generated by the front set of the five dark grey shaded nodes. In line with the flocking behavior, packets were spread in the network, but not too much, choosing a number of paths to the sink. Fig. 30(b) shows that after the activation of the bottom set of source nodes, the new flock of packets splits apart into two separate groups in order to avoid colliding with packets generated from the front

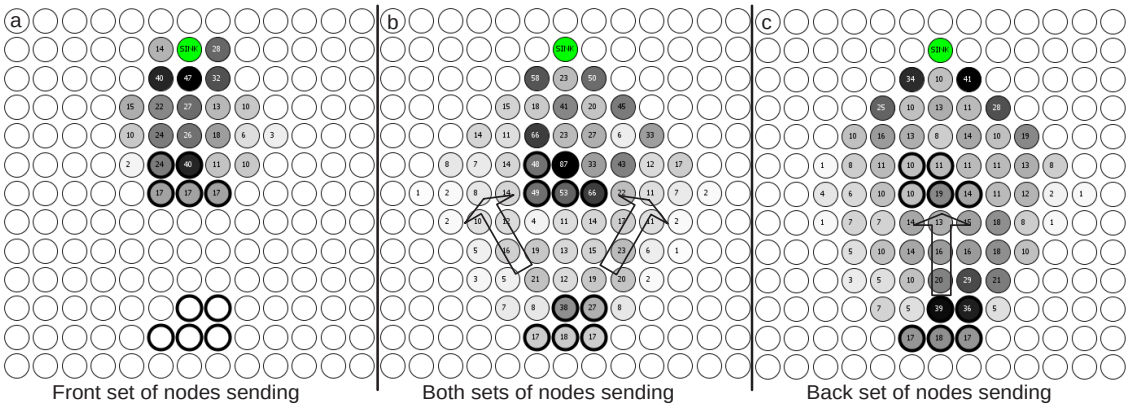


Figure 30: Emergent behavior: Visual representation of moving packets in scenario 2 (high load). The number of packets visiting each node within a 1-second time slot snapshot is indicated inside every node. Darker colors indicate higher number of packets.

set of source nodes. The direction of motion of the two sub-flocks is highlighted by transparent arrows. Furthermore, when the front set of nodes stopped transmitting, the two sub-flocks rejoined to a single, more coherent flock moving over the area that used to be a congestion region Fig. 30(c).

Fig. 31 shows the emergent behavior of moving packets before and after node failures in scenario 3. The emerging reorganization of moving packets after node failures is shown in Fig. 31(b), where the single flock of packets splits into two smaller compact groups, both of which 'fly' around the 'dead' zone. For visualization purposes, an extension to the third scenario was made. At some point after failure ($t = 70s$), the center failed node became alive, thus creating a potential new path down the middle. The adaptive motion of packets through three alternative paths to the sink is shown in Fig. 31(c). After the reactivation of the node in

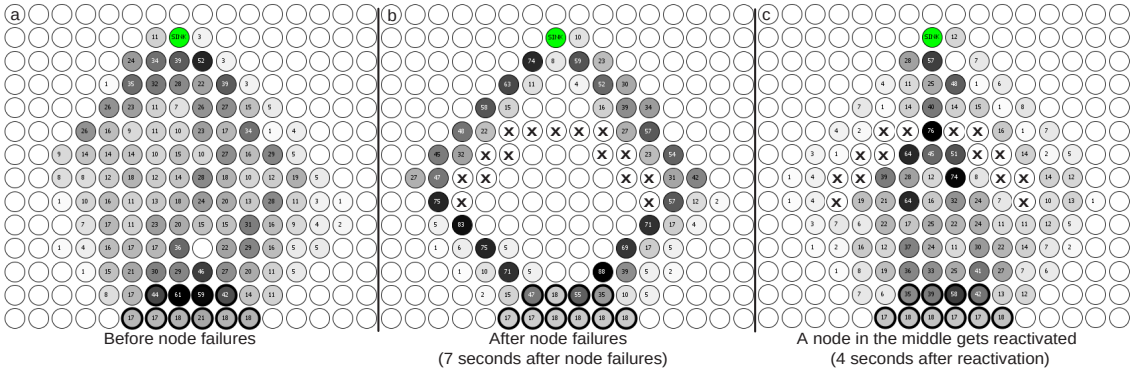


Figure 31: Emergent behavior: Visual representation of moving packets in the presence of 15 failing nodes in scenario 3 (high load): (a) before nodes fail, (b) after nodes fail, and (c) one node in the middle gets reactivated. The number of packets visiting each node within a 1-second time slot snapshot is indicated inside every node. Darker colors indicate higher number of packets.

the middle, the magnetic field lines passing through the 'hole' helped packets perceive the newly created shortest path leading to the sink. Network resources inside the 'horse-shoe' area and close to the 'hole' were incapable of accommodating the whole generated traffic load. In the case of a routing protocol forwarding packets on shortest paths, congestive phenomena would be expected to arise inside the 'horse-shoe' area towards the 'hole'. However, under the Flock-CC protocol, packets moving through the 'horse-shoe' area exhibited both repulsive as well as attractive forces, readjusting the flows both through the 'horse-shoe', as well as around it, in an adaptive dynamic fashion.

Fig. 32 depicts the obstacle avoidance behavior of packet flocks as different sets of nodes were progressively turned off. This scenario demonstrates the ability of the Flock-CC approach to find alternative paths to the sink at even more catastrophic conditions than in scenario 3. In particular, a similar 'horse-shoe' dead zone to

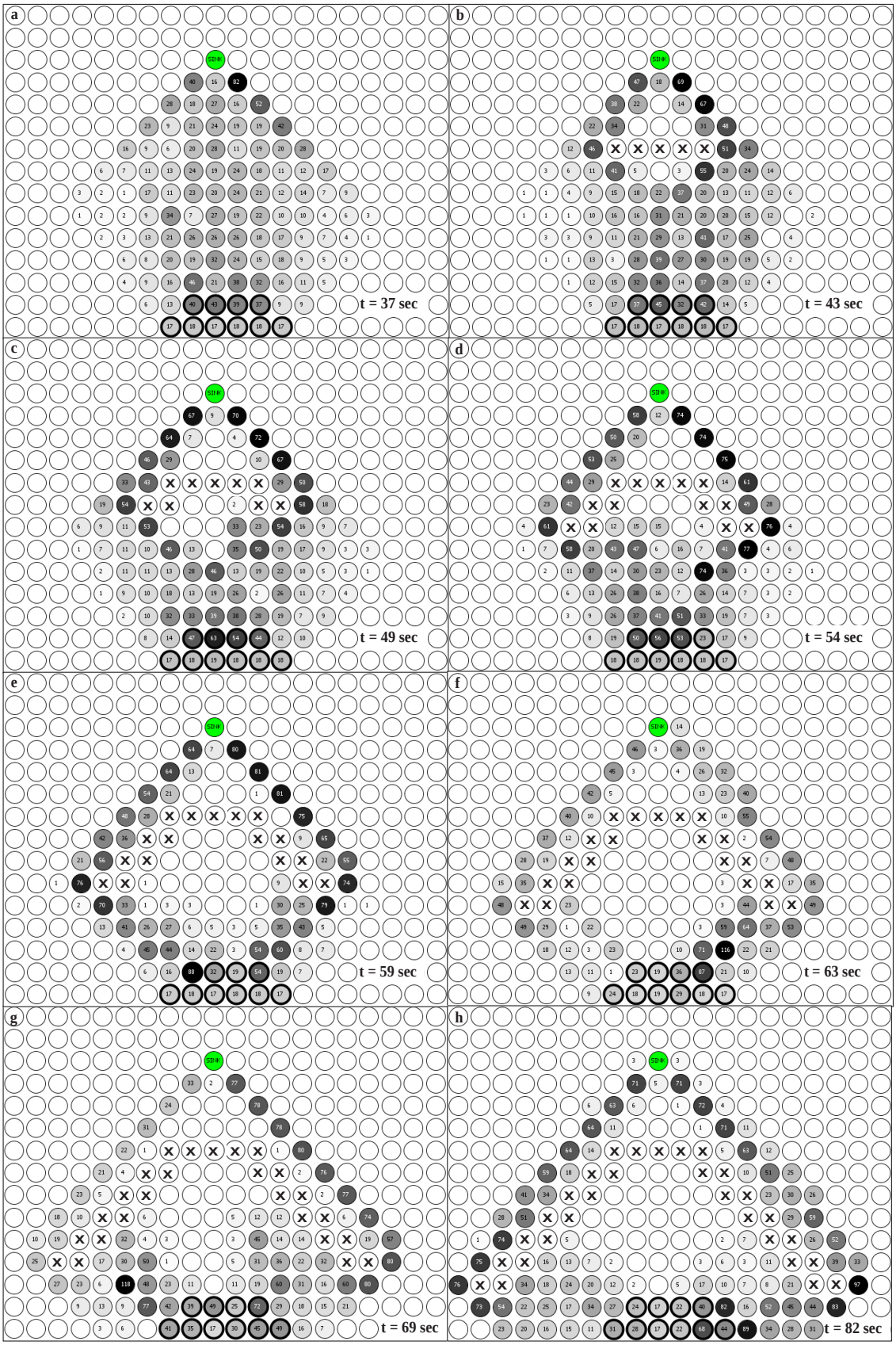


Figure 32: Emergent behavior: Visual representation of moving packets in the presence of 29 failing nodes (high load): (a) before node failures, (b)-(h) after each set of 4 failing nodes. The number of packets visiting each node within a 1-second time slot snapshot is indicated inside every node. Darker colors indicate higher number of packets.

scenario 3 was used. However, in this scenario 29 nodes were selected to gradually fail, whereas in scenario 3 shown in Fig. 31 15 nodes failed at once. In the current scenario, the first set of 5 nodes failed at $t = 40$s, and ever since then 6 sets consisting of 4 nodes each (2 nodes at each side) were failing gradually every 5 seconds until $t = 70$s. The throughput of the node located at the lower left-hand side of Fig. 32 is shown in Fig. 33, while the throughput of all active nodes in the scenario under study is shown Fig. 34.

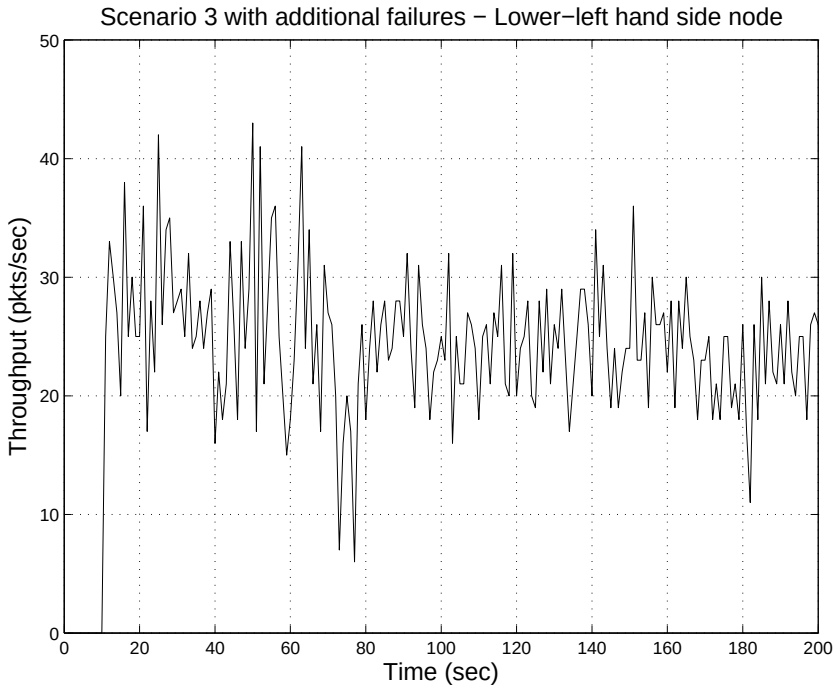


Figure 33: Throughput of the node located at the lower left-hand side of Fig. 32 measured at the sink when $\xi = 0.75$ and $T = 0.5$s, scenario 3 with additional failing nodes (high traffic load).

It can be seen that there was no steep decline in throughput as in scenario 3 (see Fig. 27) where the loss of nodes was sudden. However, as shown in Fig. 33, there was a decrease in throughput right after $t = 70$. At that moment, the final set of

nodes failed as shown in Fig. 32(h). The loss of the last set of nodes was crucial since the access of packets to the sink became quite limited (only one available path at each side left). Also, even if the throughput of active nodes in the scenario under study was fluctuating more severely than in scenario 1 (without node failures) the average number of packets received at the sink per second for both scenarios were quite similar.

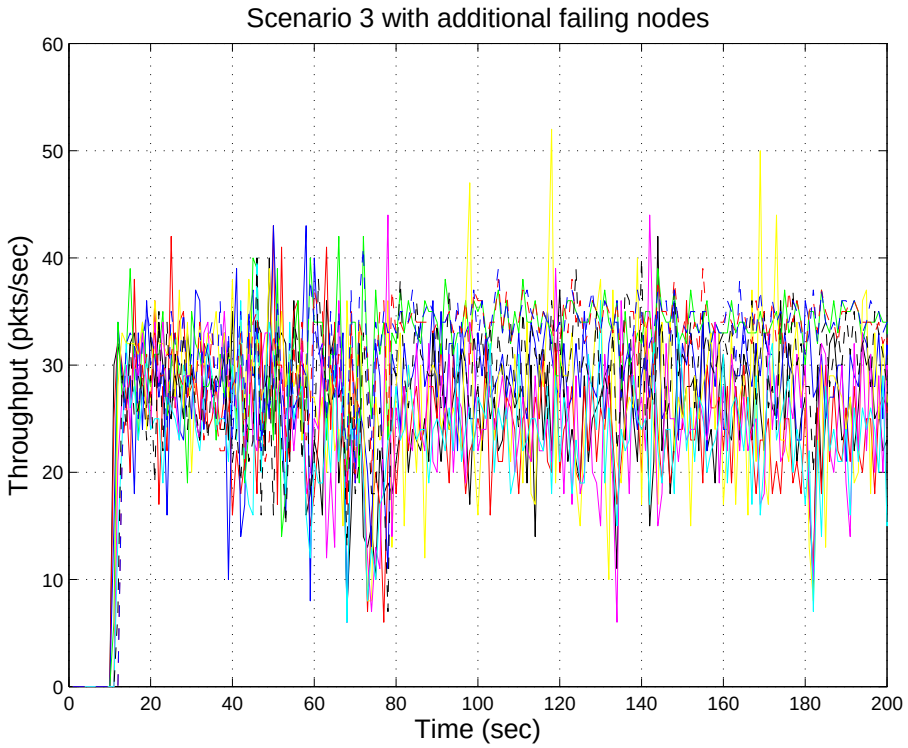


Figure 34: Throughput of each active node measured at the sink when $\xi = 0.75$ and $T = 0.5$s, scenario 3 with additional failing nodes (high traffic load).

Based on the outcomes of both Figs. 27 and 34, the Flock-CC approach (with $\xi = 0.75$ and $T = 0.5$s) can achieve reasonable (according to network resources)and similar throughput levels across failure-free and failure-prone environments. In the

presence of a sudden, catastrophic event (as in scenario 3), a small amount of time is needed for the Flock-CC approach to re-establish a new 'flight' paths to the sink and recover active nodes throughput to the same levels as before failures.

This section focused on the ability of moving packets to mimic the obstacle avoidance behavior of bird flocks. Visual representations demonstrated the *self-adaptiveness* of the proposed approach to changing traffic (Fig. 30) and network conditions (Fig. 31). In particular, it was shown that packets were dynamically moving apart to avoid queue and channel loading phenomena or node failures (thus causing traffic spreading among available paths to the sink) and moving back together once the congested or faulty situations elapsed. As a side effect, the network energy expenditure was 'fairly' shared among nodes along these paths, resulting in increased network lifetime. If packets were routed via a single path, or were spread among multiple paths in an unbalanced manner, some of these paths could progressively become over-utilized, causing an unbalanced network energy expenditure, and possibly increased contention.

Fig. 35 visualizes the packet movement in scenario 3 (at high traffic load) when one or more of the flocking behavioral rules were removed from the full Flock-CC model. The packet movement under the full Flock-CC model is also visualized Fig. 35(a)-(c) for comparative purposes. The objective here is to emphasize the need of these simple rules and demonstrate that the emerging behavior is achieved by the set of all the behavioral rules. The first experiment, Fig. 35(d)-(f), excludes only randomization (rule 4). The second experiment, Fig. 35(g)-(i), excludes only

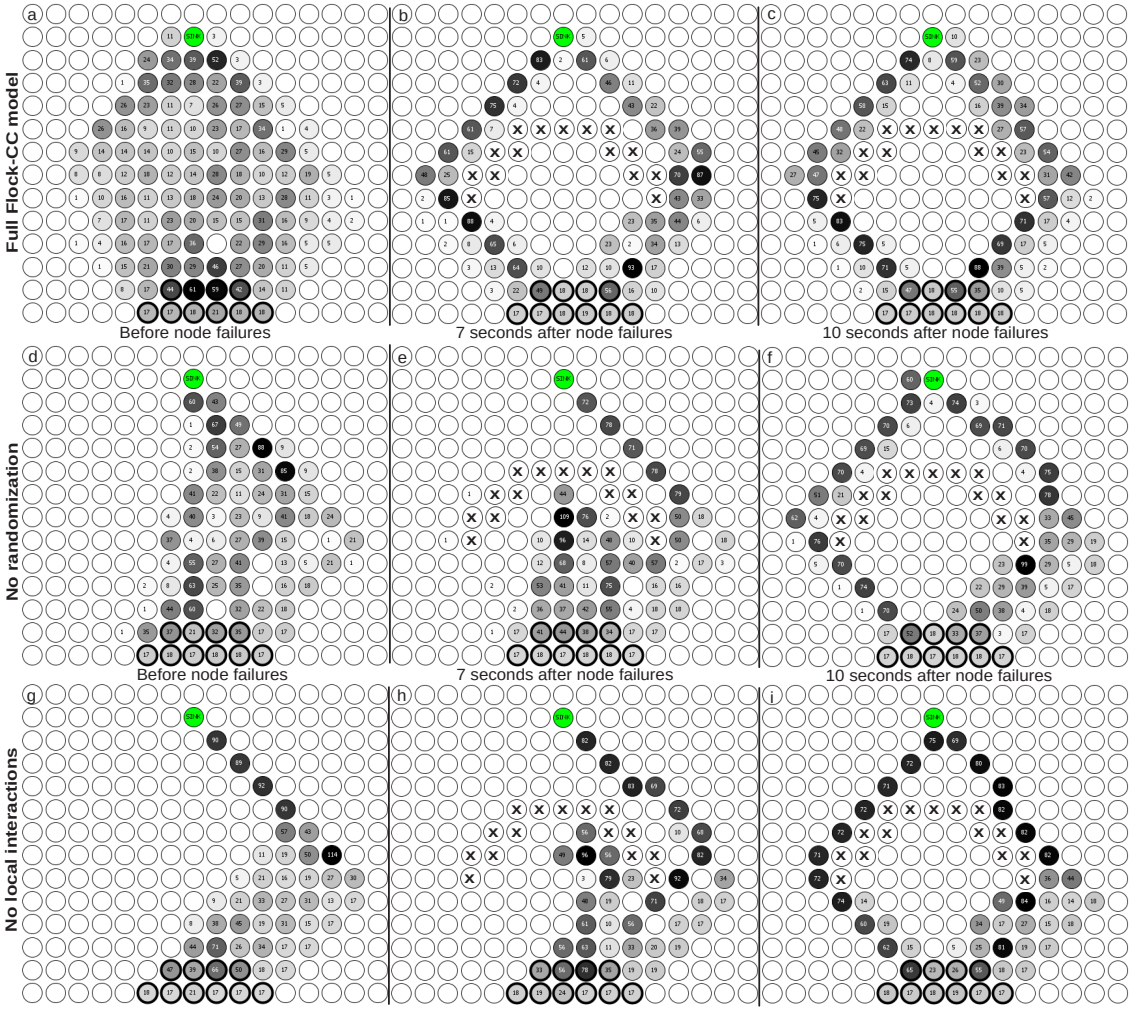


Figure 35: Lack of emergent behavior: The effect of flocking characteristics on packet flocks movement in scenario 3 (high load) when there is: (a)-(c) full Flock-CC model, (d)-(f) no randomness (rule 4), (g)-(i) no local interactions (rules 1 and 2).

local interactions i.e. repulsion and attraction forces (rules 1 and 2). The third experiment excludes both randomization and local interactions. Note that visual representations concerning the third experiment were omitted since they were almost identical to the second experiment. Figs. 35(d) and (g) illustrate the packet movement before node failures. The feature of exploration, which emerges from the *randomized selection* of new hosting nodes, allows for traffic distribution through alternative paths to the sink. This feature was apparent in Fig. 35(a), where the full Flock-CC model involving all flocking characteristics was used. On the other hand, as shown in Fig. 35(d), the exclusion of randomization eliminated packet spreading. The problem was worsened by removing only local interactions as illustrated in Fig. 35(g). Local interactions form the flock and also allow packets 'exploit' previously idle nodes (using attraction forces) and avoid congested nodes (using repulsive forces). When these forces were omitted, packet paths became too coherent, thus causing deterioration of PDR and EED. The effects on performance metrics is discussed below.

Figs. 35(e), (f) and (h), (i) illustrate the packet movement after node failures in scenarios with no randomization, or local interactions respectively. In scenario with no randomization, packets found the alternative path to the sink faster than the scenario with no local interactions. As can be seen in Fig. 35(e), 7 sec. after node failures, a few packets found the way to move at the left hand side of the 'dead zone'. This happens due to the higher tendency of packets to spread as a result of the attraction and repulsion forces. As shown in 35(b), in the full Flock-CC model,

the reaction to failures was more effective since 7 sec. after node failures the packets found both alternative paths to the sink.

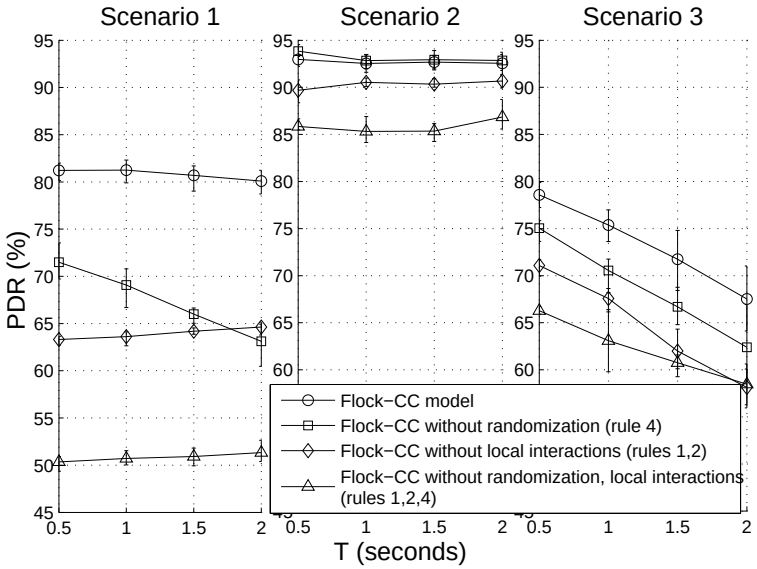


Figure 36: Emergent behavior: The effect of flocking characteristics on packet delivery ratio (PDR).

The emergent behavior was also perceived through performance evaluation metrics. Fig. 36 compares the full Flock-CC model involving all flocking characteristics against the three subvariants of the Flock-CC model described above.

Fig. 36 shows that the full Flock-CC model exhibited the highest PDR, especially for the scenarios 1 and 3. The exclusion of randomization from the Flock-CC model led to the deterioration of PDR. In scenario 1, the reduction was ranging from 9% for $T = 0.5$s to 17% for $T = 2$s, whereas in scenario 3 the reduction was from 5% to 11%. In scenario 2, where light congestion phenomena occurred, there was no significant deterioration of quality in the absence of randomization. Traffic distribution through perturbation was highly effective especially in scenarios with

large hotspot areas (scenarios 1 and 3). *Thus, the gain (low buffer overflows and collisions) from path exploration (rule* 4*), emerging from the randomized selection of new hosting nodes, is significant.*

Further PDR deterioration occurred by removing only local interactions. For scenario 1, the reduction was around 15%, for scenario 2 the reduction was from 3% to 5% and for scenario 3 the reduction was from 8% to 10%. *Clearly, the gain from local interactions among packets is even higher than what is achieved with randomization. In particular, a steep decrease in the numbers of buffer overflows and collisions emerges due to the social activity between neighboring packets (rules* 1 *and* 2*).*

As expected, the exclusion of both randomization and local interactions led to further deterioration of the PDR. Furthermore, the exclusion of the magnetic field (rule 3) and the FoV had devastating effects on PDR. Results are omitted due to the extremely low values of PDR, which even fell to $1\% - 2\%$ (scenario 1, high load).

Results concerning EED evaluations exhibited similar behavior to PDR as shown in Fig. 37.

5.1.4 Robustness in failure prone environments

Sensor nodes are prone to failures, mainly due to fabrication process problems, environmental factors (disasters), enemy attacks, and battery power depletion. The proposed approach exhibits robustness against node failures due to the inherent tendency of individuals to follow other flockmates that manoeuvre to avoid obstacles

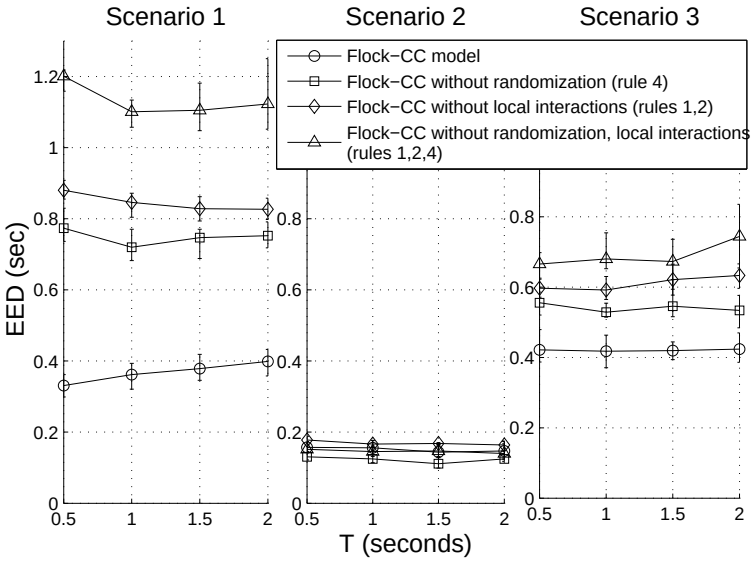


Figure 37: Emergent behavior: The effect of flocking characteristics on end-to-end delay (EED).

such as failing nodes. The third scenario was used to demonstrate the robust nature of the flock-based approach.

Fig. 31 shows network snapshots before and after node failures. It is apparent that the Flock-CC approach displayed outstanding flocking behavior in the presence of numerous node failures (selected in a horse-shoe orientation, which after failures traps packets), and exemplified all of the characteristics of a bird flock in terms of obstacle avoidance and manoeuvring around the zone of dead nodes, dynamically reforming even in the case of reactivation of nodes.

5.1.5 Scalability

The Flock-CC protocol scalability refers to the ability of the protocol to support WSNs expansion to include more nodes that might be anticipated during the initial

network design stage. This section presents the results obtained when testing Flock-CC on different sizes of networks consisting of 200, 300 and 400 nodes. In each case, a grid topology was used, having a set of 10 source nodes closely placed in the middle bottom part of the network.

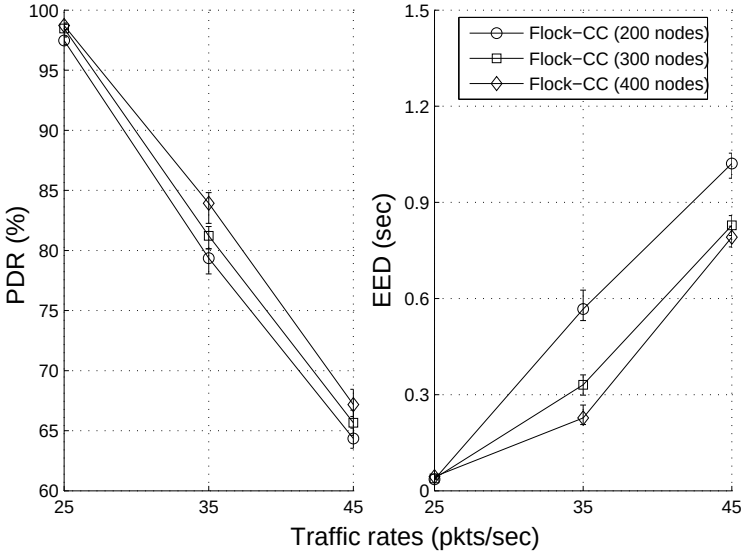


Figure 38: Scalability: The effect of network size in packet delivery ratio (PDR) and end-to-end delay (EED).

As can be seen in Fig. 38, there was a slight rise of PDR values with the increase of network size. The Flock-CC protocol is shown to perform better in large scale networks compared to small scale networks (since diversity increases). As the number of nodes in a network scales up, the amount of available resources increases and packets are able to spread widely through the network, thus minimizing packet losses, primarily due to buffer overflows. On the other hand, in small scale networks packets move in more restrictive coherent formations that increase the likelihood of buffer overflows due to the limited number of paths to the sink. Fig. 38 illustrates

the effect of network size on EED. As the network size increases (spanning same geographical area), the time needed for packets to reach the sink shortens (i.e. lower EED). This is because the increase of network resources, and as a result the increase of alternative paths to the sink, reduces the buffer occupancy (and buffer overflows) at each node. Therefore, packets face lower queueing delays and travel from source nodes to the sink at a faster pace. Finally, as expected, EED delay increases with the increase in traffic rates due to the rise in buffer occupancy at each node.

5.1.6 Comparative studies

The proposed Flock-CC protocol was quantitatively compared against NCC, CAwR and AODV [78] protocols, as well qualitatively compared against AntHocNet [12] and AntSensNet [13]. It is worth pointing out that the comparative evaluation does not include issues such as robustness and scalability. However, due to its nature Flock-CC can be expected to outperform (by far) these schemes. The characteristics of each protocol are summarized in Table 1.

Results of Figs. 39 (a)-(f) show that Flock-CC, the proposed flock-based approach, clearly outperformed NCC (no congestion control) and CAwR (congestion-aware routing) protocols in terms of both PDR and EED, for all traffic loads and scenarios. In addition, as shown in Figs. 39 (g)-(i), Flock-CC consumed less energy per delivered packet in low traffic rates (25 pkts/s) for all scenarios and slightly increased energy in higher loads. The results for the AODV protocol were omitted due to the very low performance of the protocol compared to the other protocols shown

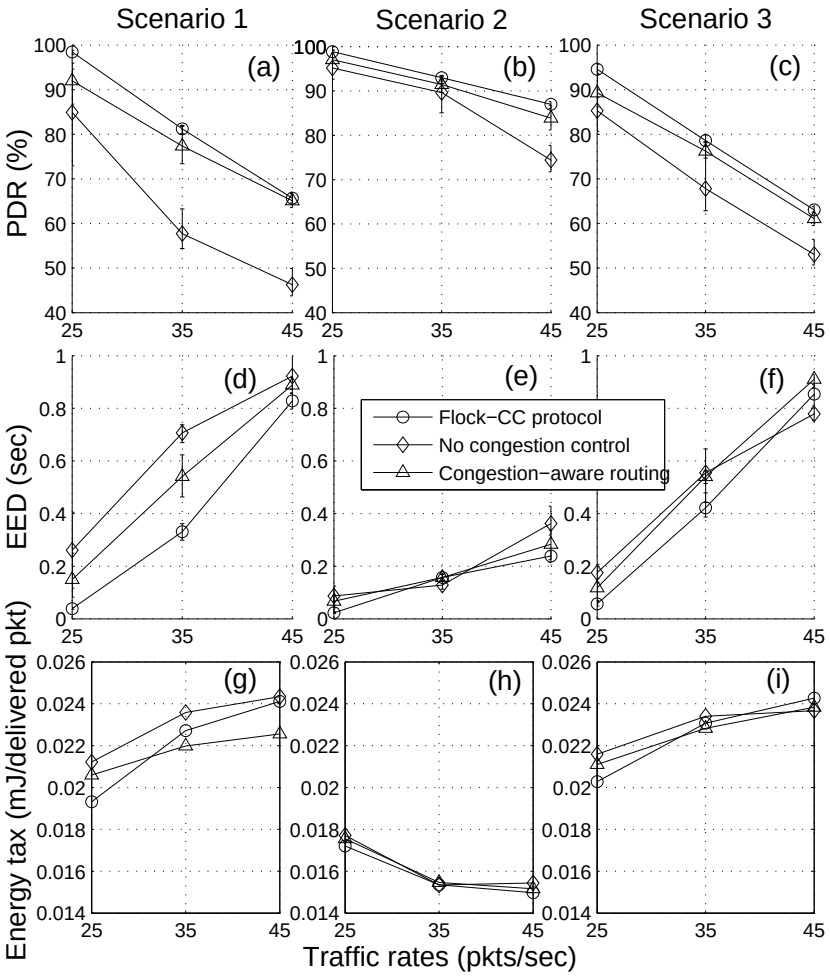


Figure 39: Comparative experiments in scenarios $1-3$ for $T = 0.5$s. The performance of AODV was considerably poorer, thus related results were omitted from this figure.

in Fig. 39. Note that AODV suffers from two limiting factors: (a) it is an end-to-end routing protocol that routes packets from a given source to a given destination through the shortest path connecting each other; (b) it is a reactive routing protocol, meaning that the path needs to be established from source on demand (when source has packets to transmit) using a series of control messages. These control messages are broadcasted in all directions, which is useless when they are sent in the opposite direction of the destination, causing high bandwidth consumption.

From the perspective of PDR, Figs. 39 (a)-(c) show that the Flock-CC approach delivered around $15\%, 23\%$ and 19% more packets for scenario 1 than the NCC protocol under low, high and extreme traffic loads respectively. The difference in PDR between scenarios 2 and 3 was smaller. Similarly, in scenario 1, the Flock-CC approach achieved 2% to 8% higher PDR (better performance in low loads with decreasing trends in extreme loads) compared to the CAwR protocol. Differences of 2% to 4% in PDR between scenarios 2 and 3 were observed.

Based on the outcomes of the comparative study, it can be argued that the controlled traffic spreading that emerged from the flocking behavior of packets allowed packets to exploit available resources on nodes involved in multiple paths to the sink, resulting in higher performance. The NCC and CAwR protocols did not allow for packet spreading among all available paths, resulting in over-utilization of some (popular) paths. Under the NCC protocol, packets were solely sent over the shortest path(s), while the CAwR protocol allowed for packet spreading after the appearance of buffer overflows. This behavior led to a high number of packet losses due to buffer

overflows. Further results showed that the Flock-CC approach exhibited extremely low buffer overflows compared to both protocols due to the traffic spreading ability of the bird flocking behavior.

In addition, Figs. 39 (d)-(f) show that the Flock-CC approach exhibited the lowest EED among the other protocols for every transmission rate and scenario, because traffic spreading prevented augmented buffer occupancies that contribute to larger queueing delays. As far as energy consumption is concerned, Figs. 39 (g)-(i) show that the Flock-CC protocol spent less energy per delivered packet in low loads compared to NCC and CAwR.

Flock-CC was also qualitatively compared against AntHocNet and AntSensNet. Due to the high degree of complexity of both AntHocNet and AntSensNet, quantitative comparative scenarios are left for future work. Below, a detailed discussion into the insights of these 2 protocols is given, revealing the difficulties of implementing accurately these approaches.

According to the researchers, AntHocNet showed better performance compared with AODV in terms of data delivery ratio and end-to-end delay [12]. Also, simulation results show that the performance of AntSensNet outperforms AODV in terms of delivery ratio, end-to-end delay and routing overhead [13].

Both AntHocNet and AntSenseNet are quite complicated protocols involving a large number of parameters and functions (see Table 2 and Table 3). These parameters have to be tuned for a variety of network and traffic conditions since they

can be sensitive to the environment. On the other hand, the Flock-CC approach is quite simple involving only two parameter and one equation (desirability function).

Both protocols are able to find routes to a possible destination in a proactive[3] and reactive[4] manner using two kinds of control packets (forward and backward ants). Re active forward ants are sent by the source node to find multiple paths towards the destination node. Backward ants are used to actually setup the route. While the data session is open, paths are monitored, maintained and improved proactively using different agents, called proactive forward ants. The protocols react to link failures with either a local route repair or by warning preceding nodes on the paths.

The overhead generated by the ants (i.e. the control packets moving back and forward into the network) is quite high for both AntHocNet and AntSensNet. During the route discovery process, several ants leave their node source, aiming for their neighbors, each one with the task of finding a route, meaning that sensor nodes must communicate with one another and the routing table of each node must contain the identification of all sensor nodes in the neighborhood as well as their corresponding levels of pheromone left on the trail. As the number of nodes grows, the number of agents required to establish the routing infrastructure may explode [90]. The high packet overhead causes scalability problems in AntHocNet and the performance of

[3]In proactive routing, while a data session is in progress, paths are probed, maintained, and improved proactively.

[4]In reactive routing, routes are set up when needed, not before. Once routes are set up, data packets are sent stochastically over the different paths using a pheromone table placed in each router.

the protocol degrades as the number of nodes in the MANET increases. AntSensNet overcomes the overhead explosion and reaches scalability using a hierarchical routing approach involving clusters. However, clustering formation assumes special roles in the network (e.g. clusterheads), while additional mechanisms are needed for maintaining and re-assigning roles. Also, areas around clusterheads may progressively become collision hot spots and deteriorate congestion. In Flock-CC protocol, the overhead is lighter since each node broadcasts periodically a control packet only. Also, the Flock-CC protocol was shown to be scalable with the increase of sensor nodes without involving any added complexity like clustering.

In ant-based approaches, the problem of overhead is worsened since reactive forward ants (packets used for exploration) store the full array of nodes that they have visited on their way to the destination. In this way, a large amount of information is progressively added to these packets, thus increasing the probability of collision at considerably high levels. Also, the amount of energy needed to transmit or receive a packet increases. Finally, the effect of packet loss on robustness is expected to be non-negligible. On the other hand, in Flock-CC, the size of a control packet remains always small and constant and each control packet is transmitted over only one hop.

Furthermore, both AntHocNet and AntSensNet necessitate large memory space to store all information used by the protocol. Indicatively, AntSensNet involves four pheromones for each traffic class. Every node x has to keep in its pheromone table the four pheromone values for each traffic class that passes through each link connecting node x and its neighboring nodes. Similarly, in AntHocNet, each node

i maintains one pheromone table T_i which is a two-dimensional table. An entry T_{ij}^d of this matrix contains information about the route from node i to destination d over neighbor j. This includes a regular pheromone value τ_{ij}^d, a virtual pheromone value ω_{ij}^d, and an average number of hops h_{ij}^d. The regular pheromone value τ_{ij}^d is an estimate of the goodness of the route from i to d over j. The virtual pheromone value ω_{ij}^d forms an alternative estimate of the goodness of the route from i to d over j. Apart from a pheromone table, each node also maintains a neighbor table N_i, a one-dimensional vector, in which node i keeps track of its neighbors. Only highly capable sensor nodes with extended memory capacities are able to store this mass of information. On the other hand, in Flock-CC, each node employs four small uni-dimensional tables that contain information about neighboring nodes only. The size of each table is equal to the number of nodes in the FoV. Table 2 and Table 3 summarize the functions and parameters employed in each of the aforementioned three approaches. The simplicity of Flock-CC is highlighted by the quite small number of functions and parameters employed in the protocol compared to AntHocNet and AntSensNet. Both AntHocNect and especially AntSensNet involve 2 and 5 times more functions and parameters than Flock-CC respectively.

In addition, AntSensNet requires modifications in the queueing policies of the underlying MAC protocol in order to accept multi-class and multi-priority traffic. On the other hand, AntHocNet does not rely on information that are available by the MAC layer protocols. AntHocNet collects gathers, maintains and updates information about paths (quality of wireless links) and destinations that are involved

in communication sessions using dedicated control packets (forward and backward ants). As mentioned before, the cost of complexity and overhead is high. Flock-CC needs only local information that can be obtained by the MAC layer protocol of each node. Flock-CC cooperates well with a large number of MAC protocols without necessitating important modifications in the MAC layer. Flock-CC needs to obtain link quality measurements and queue occupancy from the underlying MAC layer protocol. More specifically, Flock-CC needs three values from the underlying MAC protocol: (a) the number of packets in the queue at the end of each sampling period, (b) the number of packets successfully transmitted within each sampling period and (c) the number of total transmission attempts within each sampling period. These information is available at any MAC layer protocol. The overwhelming majority of MAC protocol implementations do not make these information available to higher layers. Thus, dedicated functions are needed to be implemented in the MAC layer. It is worth noting that the IEEE 802.11k group (Radio Resource Measurement) is currently developing a standard which is intended to improve the provision of data from the physical and medium access layers by defining a series of measurement requests and reports that can be used in the upper layers to carry different radio resource management mechanisms. The current draft version is 9.0 [91].

All the aforementioned issues make the implementation of both AntHocNet and AntSensNet quite complicated, difficult and time-consuming process.

The following table summarizes the differences among the three nature-inspired protocols Flock-CC, AntHocNet [12] and AntSensNet [13]:

Table 2: Comparison of features among Flock-CC, AntHocNet [12] and AntSensNet [13] (1).

	Flock-CC	*AntHocNet [12]*	*AntSensNet [13]*
Protocol functions	hop distance: h_i channel loading: s_{nm}^{norm} queue occupancy: q_{nm}^{norm} desirability: D_{nm}	average number of hops: h_{ij}^{d} regular pheromone value: τ_{ij}^{d} virtual pheromone value: ω_{ij}^{d} probability i chooses n (ant packets): P_{in}^{d} probability i chooses n (data packets): P_{nd} reported pheromone value: υ_{i}^{d} bootstrapped pheromone value: κ_{ji}^{d} cost hopping from i to j: c_{j}^{i}	maximum packet loss rate: $pl(n)$ available memory: $ma(n)$ queueing delay: $dl(n)$ normalized remaining energy: $re(n)$ unicast path delay: $delay(P)$ unicast path packet loss: $packetloss(P)$ unicast path energy: $energy(P)$ unicast path memory: $memory(P)$ objective function: $f(P_C)$ clustering pheromone value: $\Phi_c(n)$ probability of choosing neighbor (ant): $prop_c(j)$ energy pheromone value: $e_i^k(j)$ delay pheromone value: $\delta_i^k(j)$ packet loss pheromone value: $\epsilon_i^k(j)$ available memory pheromone value: $\mu_i^k(j)$ normalized pheromone value: $\Psi_i^k(j)$ probability of moving from CH to i : $P_i^k(j)$ normalized energy probability: $p_{e,i}^k(j)$ normalized delay probability: $p_{\delta,i}^k(j)$ normalized packet loss probability: $p_{\epsilon,i}^k(j)$ normalized available memory probability: $p_{\mu,i}^k(j)$

Table 3: Comparison of features among Flock-CC, AntHocNet [12] and AntSensNet [13] (2).

Elements	*Flock-CC*	*AntHocNet [12]*	*AntSensNet [13]*
Protocol parameters	spreading factor: ξ sampling period: T	adaptation to new information: α speed of adaptation: γ exploratory ant behavior control: $\beta 1$ exploratory ant character control: $\beta 2$ forwarding control: $\beta 3$	delay weight factor: γ_C^d packet loss rate: γ_C^p residual energy ratio: γ_C^r available memory for global QoS: γ_C^m memory capacity importance: α energy importance: β energy pheromone improvement factor: ρ_e delay pheromone improvement factor: ρ_δ packet loss pheromone improvement factor: ρ_ϵ memory pheromone improvement factor: ρ_μ
Types of control messages	Hello	Hello Forward Ant Backward Ant	Forward Ant, Backward Ant, Maintenance Ant Data Ant
Topology cost for maintenance	None	None	Clustering formation and maintenance mechanisms
Control packet size	Constant	proportional to the number of hops from source to sink	proportional to the number of hops from source to sink
Modifications in MAC protocol	A few	None	New queueing policies

5.2 The Lotka Volterra based congestion control approach

This section evaluates the performance of the LVCC model and discusses the effectiveness of the model in preventing congestion by mimicking the species competition in nature. More specifically, control system type simulations (through Matlab [92]) and realistic network simulations (using NS2 network simulator [14]) were conducted to show the basic features of the proposed bio-inspired mechanism such as *graceful performance degradation, self-adaptiveness, scalability and fairness*. In addition, evaluation studies investigate how parameters affect the performance of the proposed mechanism in terms of *stability and convergence* and provide effective parameter setting on the basis of congestion-oriented metrics.

5.2.1 Analytical results: The basis

Based on analytical results of Chapter 4 about α and β, the calculated rates of all flows converge to a global and asymptotically stable solution when $\beta > \alpha$, and $\alpha > 1$ for avoiding buffer overflows. Note that there is no upper limitation on β but as it becomes larger, the steady state traffic rate (Eq. 11) decreases. In this case, each active node will be limited to transmit data at a lower rate leading to lower quality of the received streams at the sink. As far as r is concerned, the system of Eq. 4 has a stable equilibrium point for any value of $r > 0$ [6], [68]. An upper bound for r is not analytically known, thus will be experimentally explored. The mathematical analysis of the proposed model gives a general understanding of the system's behavior on the basis of stability as function of the parameters α and β.

However, the complexity of WSNs necessitates simulation evaluation using plausible scenarios that cannot be formally tested. The analytical study serves as the basis for the simulations.

5.2.2 Simulation studies: The step further

In order to supplement the analytical results, some simulation experiments were conducted both in Matlab [92] (theoretical model analysis scenarios) and in NS2 [14] (realistic network scenarios).

Topology: A wireless sensor network consisting of 25 wireless nodes was considered having nodes being deployed in a cluster-based topology (Fig. 40). The proposed approach can be efficiently and effectively used on top of routing or MAC protocols that create small depth (< 4) cluster/treebased logical topologies over any physical topology. However, a detailed study of such protocols is beyond the scope of this study. In this study, a dedicated routing protocol that creates the underlined topology was assumed. This type of topology was used so as to better understand and evaluate the behavior of the LV-based mechanism. The grey-shaded area represents a collision domain. For example, the nodes of cluster 1 (nodes 5, 6, 7, 8, and 9) will perceive each other's transmissions. Also nodes 1, 2, 3, and 4 can reach the sink.

Theoretical model analysis using control system type simulations: The validity of analytical results in complex scenarios that could not be formally tested was further investigated in Matlab. It was assumed that nodes 5, 6, 10, 14, 16 and 20

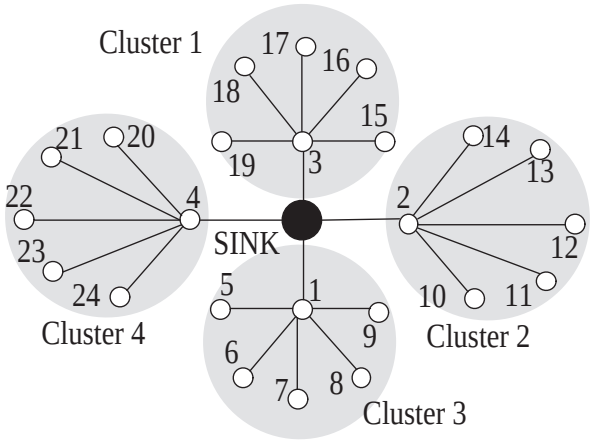


Figure 40: Evaluation cluster-based topology of 25 nodes (all links are wireless).

were activated at $1T$, $150T$, $300T$, $450T$, $600T$ and $900T$ respectively. Node 14 was deactivated at $750T$. Each node buffer size was set to $K = 35$ KB.

Realistic network simulations: In addition, the proposed mechanism was evaluated in a realistic static and failure-free network environment, using a series of representative network operation scenarios under NS2 networking simulator. The two-ray ground radio propagation model was used in all experiments. The buffer capacity of each node was set to 35 KB. The time period T between successive evaluations of the calculated rate of each SN, as well as the time between backpressure control packets was set to 1 s. The selection of 1 s is guided by the desire to maintain responsiveness to changes in the network state and to avoid overwhelming the network with control packets. The CSMA-based IEEE 802.11 MAC protocol with 1 Mbps transmission rate and an exponential backoff policy was adopted. Table 4 summarizes all scenarios evaluated in NS2. In each scenario, different sets of nodes were activated.

Table 4: Description of scenarios in NS2

Scenario	No. of active nodes	Active nodes
1	3	$5, 6, 10$
2	5	$5, 6, 10, 13, 14$
3	7	$5, 6, 10, 12, 13, 14, 21$
4	10	$5, 6, 10, 11, 12, 13, 14, 18, 21, 24$

Based on the LV competition model, each node is able to calculate its transmission rate i.e., the number of bytes it can send per time unit. In realistic scenarios, it was assumed that each node will transmit in one of 5 different levels namely, $1, 2, 4, 6,$ and 8 Kbytes per $T = 1$ s, starting from 1 Kbytes/T (i.e. 8 Kbps). Each node can increase its flow (or stream) rate to an upper level rate only when the calculated transmission rate exceeds the specific upper level rate. The calculated transmission rate (in bytes/T) is given by Eq. 19, reproduced here for easy reference:

$$x_i((k+1)T) = \frac{w(kT)x_i(kT)}{\beta x_i(kT) + [w(kT) - \beta x_i(kT)]\, e^{-\frac{w(kT)r}{K}T}}, \quad w(kT) = K - \alpha C_i(kT). \quad (19)$$

For example, if $x_i = 1.5$ Kbytes/T, node i transmits at 1 Kbyte/T. As x_i ranges from 2 to less than 4 Kbytes/T, node i transmits at 2 Kbytes/T. Similarly, there should be a transition from the current level rate to a lower level rate when the calculated transmission rate falls below the current level rate but is above the lower level rate. For example, if node i transmits at 2 Kbytes/T and x_i falls below 2 Kbytes/T, then node i can transmit at 1 Kbyte/T.

Performance metrics: Two common performance metrics for congestion control approaches were taken into account: the packet delivery ratio (PDR) and the end-to-end delay (EED). The definitions of both performance metrics are given in Section 5.1.1.5.

5.2.3 Verification of stability and convergence time through control system type simulations

Matlab [92] is a technical computing software that can be used for control system type simulations. In these simulations, realistic network conditions such as queueing delays and wireless channel collisions were not considered.

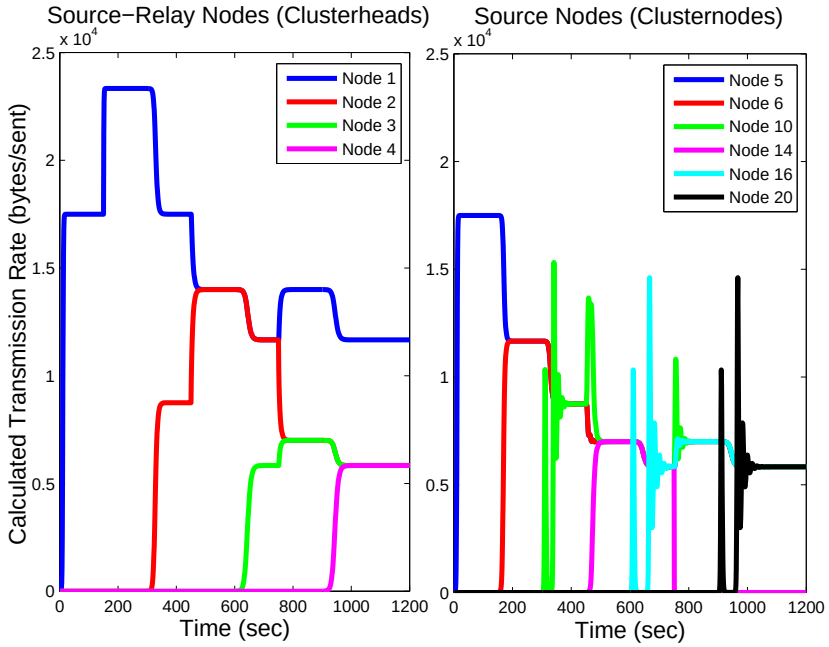


Figure 41: Calculated transmission rate (bytes/s) when $\alpha = 1$, $\beta = 2$, $r = 1$.

Initially, α and r were set equal to 1 while the value of β varied. Figs. 41-43 illustrate the results obtained using Matlab. Fig. 41 depicts the calculated number of bytes that can be sent per T from each active node when $\beta = 2$. Bear in mind that low α and β values result in high calculated transmission rates at equilibrium, x^*, as evaluated by Eq. 11. As can be observed, the system was able to re-converge to a new stable point after each change in network state (node

activation), with some fluctuations in calculated sending rates exhibited by flows initiated from nodes 10, 16 and 20. This behavior[5] is attributed to the fact that the buffers of nodes involved in the path between active nodes were highly loaded (since traffic flow rates were allowed to converge at high equilibrium values). Thus, with the activation of a new node, the increase of traffic injected into the network could not be smoothly accommodated by network's resources. Also, some fluctuations occurred when the flow of node 14 was deactivated. However, buffer overflows never occurred since the buffer overflow avoidance condition ($\alpha[n-1]+\beta \geq n$) was satisfied. On the other hand, high traffic load injection into the network may lead to wireless channel capacity saturation, a phenomenon that was apparent in realistic network simulations.

When β increased to 4 (Fig. 42) all flows became almost well-behaved while some very small fluctuations occurred after changes in the number of active nodes. Recall that the increase of β resulted in convergence of calculated rates at smaller equilibrium values x^*. As a result, the buffers within the network were not highly loaded. Hence, the increase of the traffic injected into the network was conveniently accommodated by network resources, while smooth converging behavior of the calculated transmission rates was preserved. Even though there is no analytical upper bound for β value, as β increases, the incoming traffic load can be conveniently accommodated but the quality of the received data at the sink may be reduced. It can be argued that the best setting for parameter β would be the lowest value

[5]In control theoretic terms, this condition would indicate that some damping is required. When β is increased (e.g. to 4), this condition is illustrated, but with reduced steady state transmission rates, as theory predicts

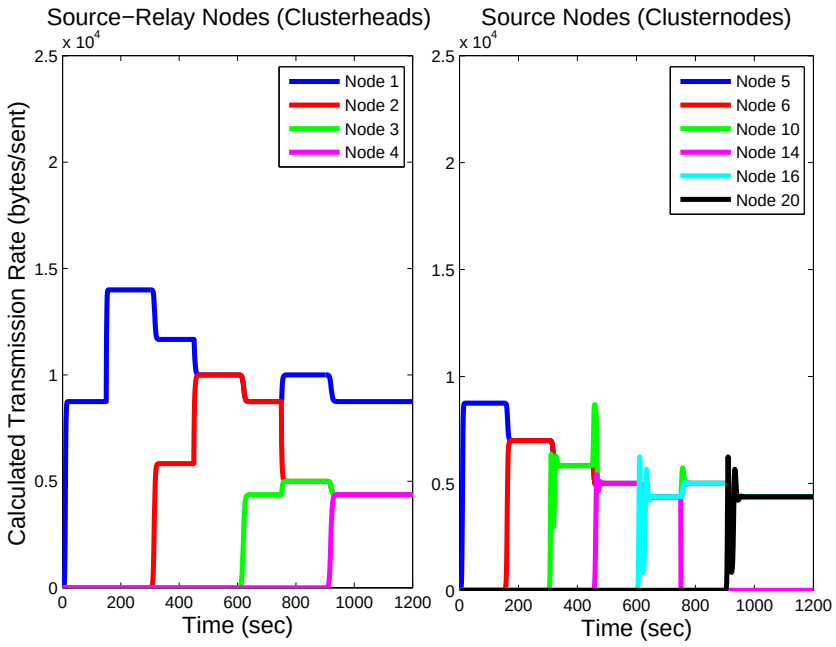


Figure 42: Calculated transmission rate (bytes/s) when $\alpha = 1$, $\beta = 4$, $r = 1$.

that ensures stability and high calculated transmission rates at equilibrium (and thus, high quality), without causing wireless channel capacity saturation and buffer overflows. The upper bound for β is further explored in realistic network scenarios.

The role of parameter α is discussed on the basis of Fig. 43. In this scenario, parameters α and β were set to 3 and 4 respectively. Based on both buffer overflow avoidance and stability conditions, parameter α is lower bounded by 1 and upper bounded by β respectively ($1 < \alpha < \beta$). As can be seen, large oscillations were observed at source nodes (Fig. 43(b)) because the system was close to the stability limits. In addition, parameter α was found to be proportional to convergence time. Thus, fast convergence to the stable solution requires α to be close to 1 rather

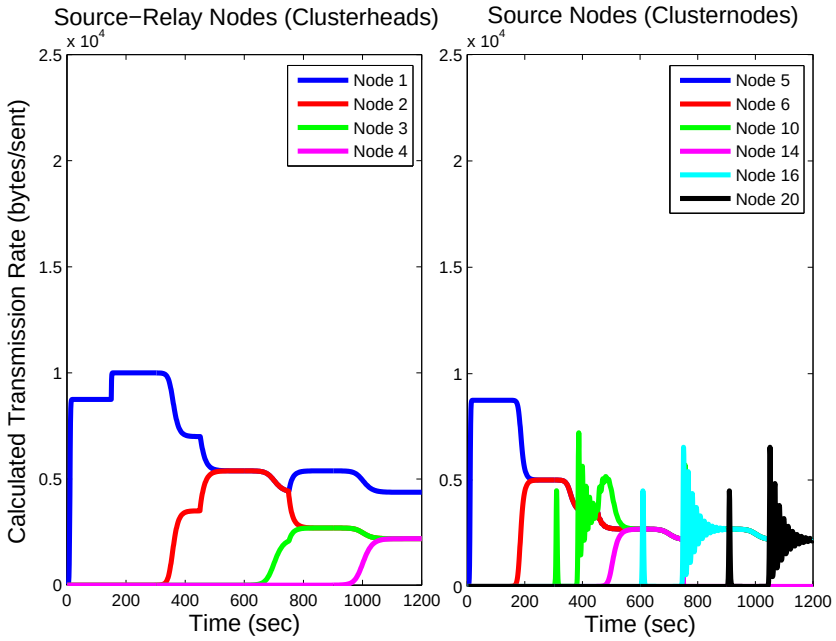


Figure 43: Calculated transmission rate (bytes/s) when $\alpha = 3$, $\beta = 4$, $r = 1$.

than close to β (i.e. far from stability limits). This analytical finding is supported by Fig. 43 which illustrates the slow response of the system towards convergence when α was close β. On the other hand, low α values result in high calculated transmission rates at equilibrium that may not be accommodated by the underlying wireless medium. This issue as well as the influence of α on system performance were further investigated in realistic network scenarios.

In all the previous scenarios, the parameter r was set to 1. Further matlab simulation studies were carried out in order to study the influence of r on stability. Recall that r was analytically found to be inversely proportional to convergence time, i.e. how fast or slow the system converges to the stable solution. Simulation results showed the value of r cannot grow unboundedly in order to achieve fast

convergence. The value of r was tested across quite a large number of combinations of α and β values. Results showed that the calculated flow transmission rates were able to converge for every combination of α and β when $r \leq 2$.

5.2.4 Parameter setting using NS2-based realistic network scenarios

In this section, the impact of parameters α, β and r on a realistic network environment is investigated. Each scenario, concerning different combinations of α, β and r values, was executed 10 times and the average values of metrics over all scenarios are presented below. The values of each parameter were chosen to be $4 \leq \beta \leq 7$, $1 \leq \alpha \leq 4$ (in order to satisfy the conditions of stability and buffer overflow avoidance), and $0.5 \leq r \leq 2$. Initially, parameter r was set to 1.

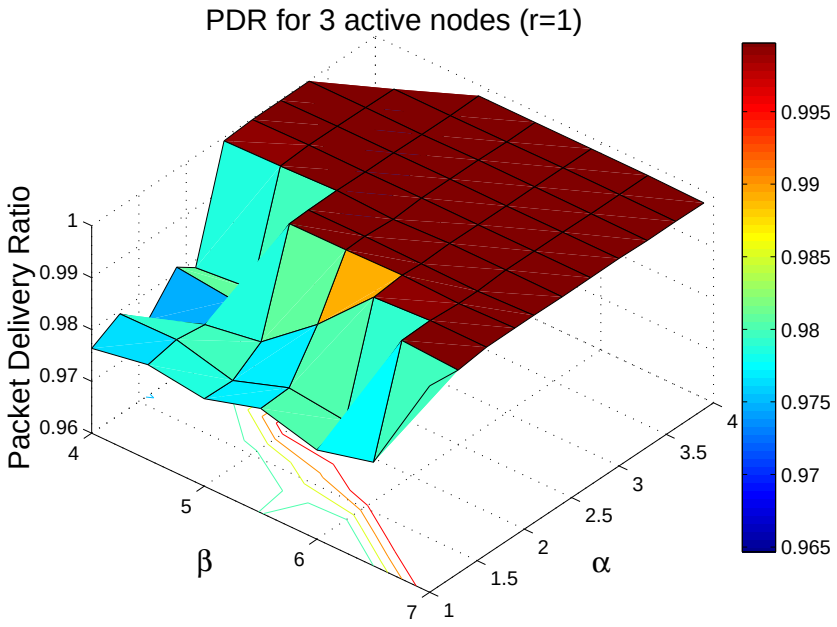


Figure 44: Packet Delivery Ratio for 3 active nodes ($r = 1$).

Fig. 44 illustrates the impact of α and β on packet delivery ratio (PDR) when 3 nodes were active. It can be observed that the PDR for all active nodes was close to 1 (i.e. the sink received almost all packets sent from all active nodes) for the majority of β and α values. More specifically, for high values of β as for example $6 \leq \beta \leq 7$, high PDR was achieved for almost all values of α. Similarly, high PDR (close to 1) was achieved for lower β values $5 \leq \beta \leq 6$ when $2 \leq \alpha \leq 4$, and for $4 \leq \beta \leq 5$ when $2 \leq \alpha \leq 3$. Realistic network simulation results validated control system type simulations. In particular, the decrease in PDR perceived for low values of α was mainly attributed to the increase in calculated transmission rates at equilibrium. Thus, the increase of traffic load injection into the network provoked wireless channel contention leading to packet loss. In addition, a sharp decrease in PDR was observed when the stability condition was threatened, as for example for $3.5 \leq \alpha \leq 4$ and $\beta = 4$.

Fig. 45 presents the PDR for 10 active nodes. The highest PDR (≈ 0.9) was obtained for $6 \leq \beta \leq 7$ and $1.8 \leq \alpha \leq 2.1$. In addition, low delay values ($\approx 10\mu s$) were achieved when α was set between 1.8 and 2.1, while β was ranging between 6 and 7.

Fig. 46 takes a closer look at the behavior of active flows under changing parameter values. The aim is to reveal how the violation of conditions for stability and buffer overflow avoidance could impact smooth network operation. In particular, Fig. 46(a) shows the calculated transmission rates and Fig. 46(b) illustrates the stream throughput measured at the sink for the 3 active nodes for 3 combinations

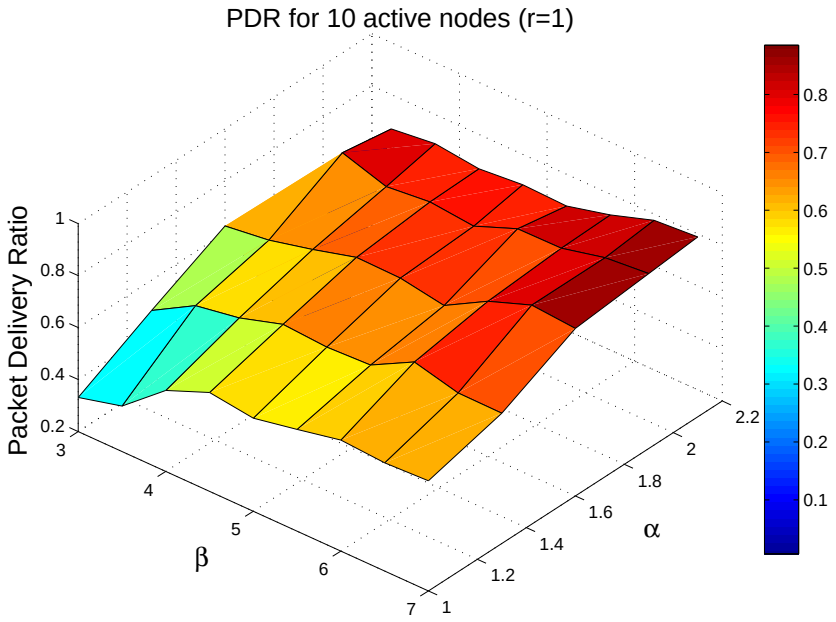


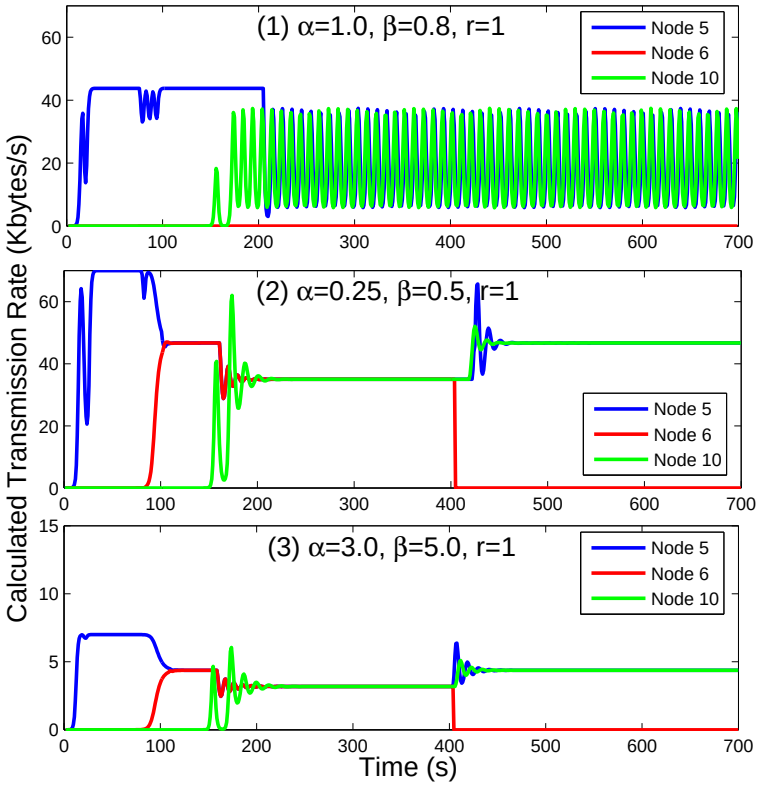
Figure 45: Packet Delivery Ratio for 10 active nodes ($r = 1$).

of parameter values. Table 5 refers to the validation of stability and buffer overflow avoidance conditions for the scenarios of Fig. 46.

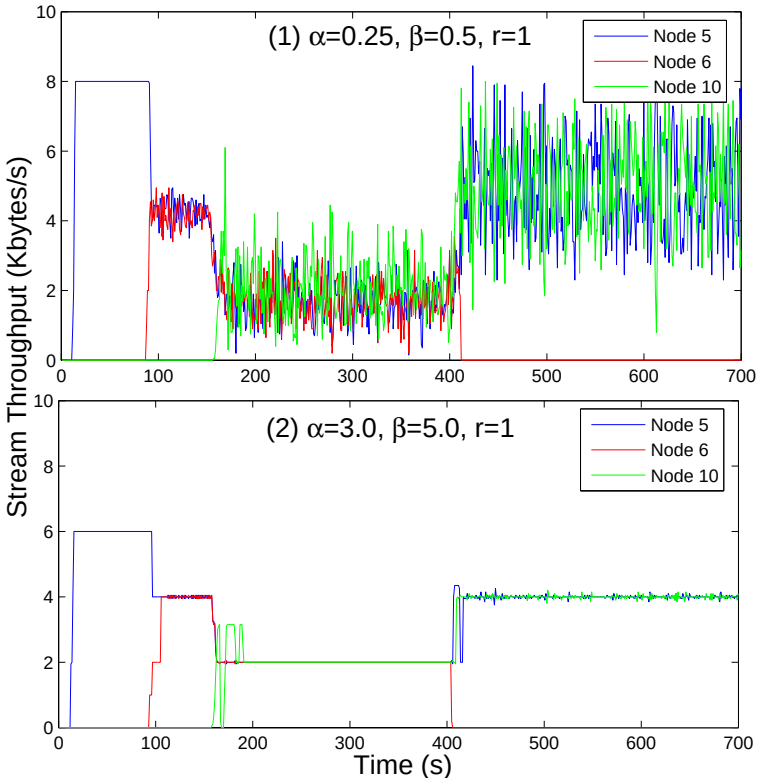
Table 5: Validation of stability and buffer overflow avoidance conditions for scenarios of Fig. 46

α	β	$\beta > \alpha$	$\alpha\,[n-1] + \beta \geq n$	x^* (Kbytes/s) when all active
1.0	0.8	X	$2.8 \geq 3$ X	–
0.25	0.5	$\surd$	$1 \geq 3$ X	35
3.0	5.0	$\surd$	$11 \geq 3$ $\surd$	3.18

As can be seen in Table 5, when $\alpha = 1$ and $\beta = 0.8$, neither stability nor buffer overflow avoidance conditions were satisfied. In these scenarios, nodes $5, 6$ and 10 started transmitting at $t = 4, 74$ and 144 s respectively, while node 6 stopped transmitting at $t = 404$ s. Due to the violation of the first condition, only the flows of nodes 5 and 10 survived while the flow of node 6 became extinct. In addition, the



(a) Calculated transmission rates



(b) Stream throughput measured at the sink

Figure 46: Scenario with 3 active nodes ($r = 1$).

calculated transmission rates of the survived active nodes exhibited cycle instability as shown in Fig. 46(a)(1). The phase plane of this scenario shown in Fig. 47 illustrates the oscillatory nature of the population of the two survived flows. In addition, due to the violation of the second condition, the summation of calculated rates of the survived active nodes was greater than the buffer capacity of each node (35 Kbytes), thus leading to buffer overflows.

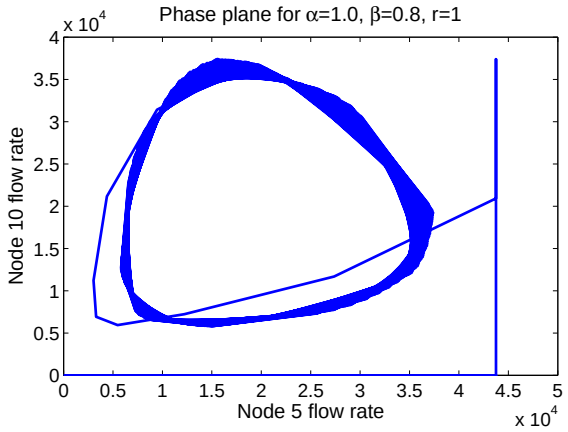


Figure 47: Phase plane for survived nodes: Cycle instability in calculated transmission rates.

As illustrated in Fig. 46(a)(2), when $\alpha = 0.25$ and $\beta = 0.5$, the stability condition was satisfied whereas the buffer overflow avoidance condition was violated. Fig. 46(a)(2) shows the calculated transmission rates after convergence, x^*, for each active node. As can be seen, the calculated transmission rate of each node is higher than or equal to 35 Kbytes/s. However, nodes were not actually transmitting at such high rates but at the highest level of 8 Kbytes/s throughout the scenario duration. Even though the buffer capacities within the network could accommodate the generated traffic load, collisions at the wireless channel led to packet loss. As a result,

the stream throughput for each active node measured at the sink was fluctuating as shown in Fig. 46(b)(1).

On the other hand, as shown in Fig. 46(a)(3), when $\beta = 5$ and $\alpha = 3$ none of the conditions were violated, while the calculated transmission rates were kept at lower values. Thus, each node was transmitting at the highest allowed predetermined transmission rate (see 46(b)(2)) without causing packet loss. Due to the low traffic load injected into the network in the presence of 3 active nodes, the mean end-to-end (EED) delay was kept below $4\mu s$.

The results of Fig. 46(a) show that for proper parameter selection, stability can be preserved under dynamically changing traffic injection in the network caused by variation in the number of active nodes. However, note that earlier results indicate the need to carefully tune the parameters according to different network/traffic conditions. For example, as seen before, for high number of active flows the setting of parameters is very sensitive. Thus, an adaptive scheme for changing parameter value according to network and traffic conditions can be proposed.

In addition, Fig. 48 depicts the influence of parameters β and α on EED when 10 active nodes were involved. Low delay values ($\approx 10\mu s$) were achieved when α was set between 1.8 and 2.1, while β was ranging between 6 and 7.

Analytical evaluations suggested that high values of r can contribute to fast convergence to the stable equilibrium solution. However, theoretical model analysis of complex scenarios in Matlab showed that network stability was achieved for $r \leq 2$. Increasingly, realistic network experiments in NS2 showed that for $r << 1$ (e.g.

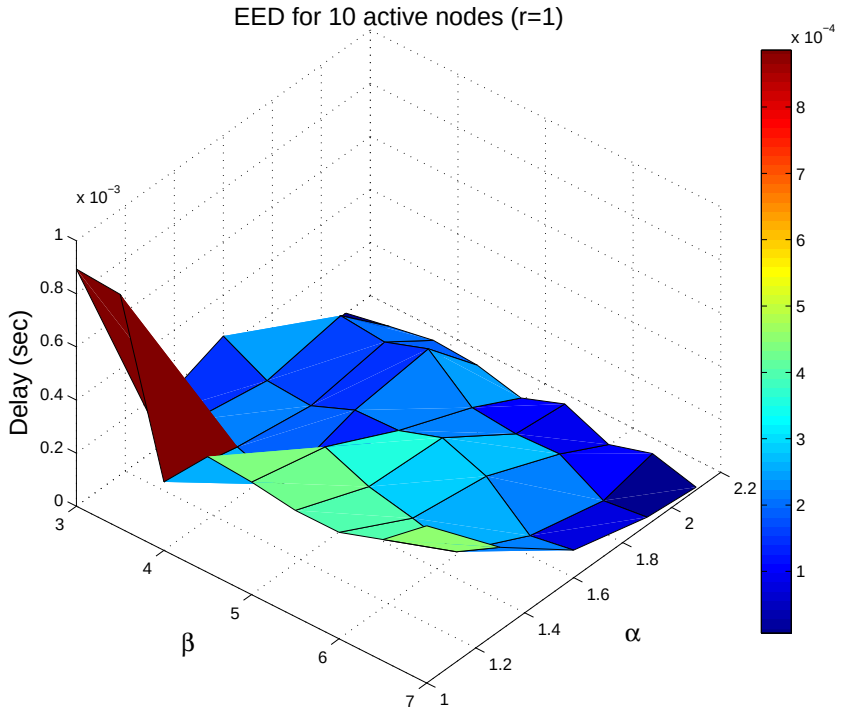


Figure 48: End-to-end delay for the fourth scenario of Table 4 involving 10 active nodes ($r = 1$).

$r = 0.5$ in Fig. 49), the calculated transmission rates of active nodes were not able to converge. On the other hand, convergence of calculated transmission rates was achieved for $r = 1$. Extensive simulation results showed that the value of r must be kept between 1 and 2 (included) in order to preserve system stability for all combinations of α and β values, regardless of the number of active nodes.

Table 6 presents the combinations of α and β values that achieved the highest PDR for different number of active nodes. The parameter r was set to 1 in order to preserve smooth flow rate regulation. The last column of Table 6 shows results obtained using 2 Mbps transmission rate at the MAC layer. It is worth pointing out

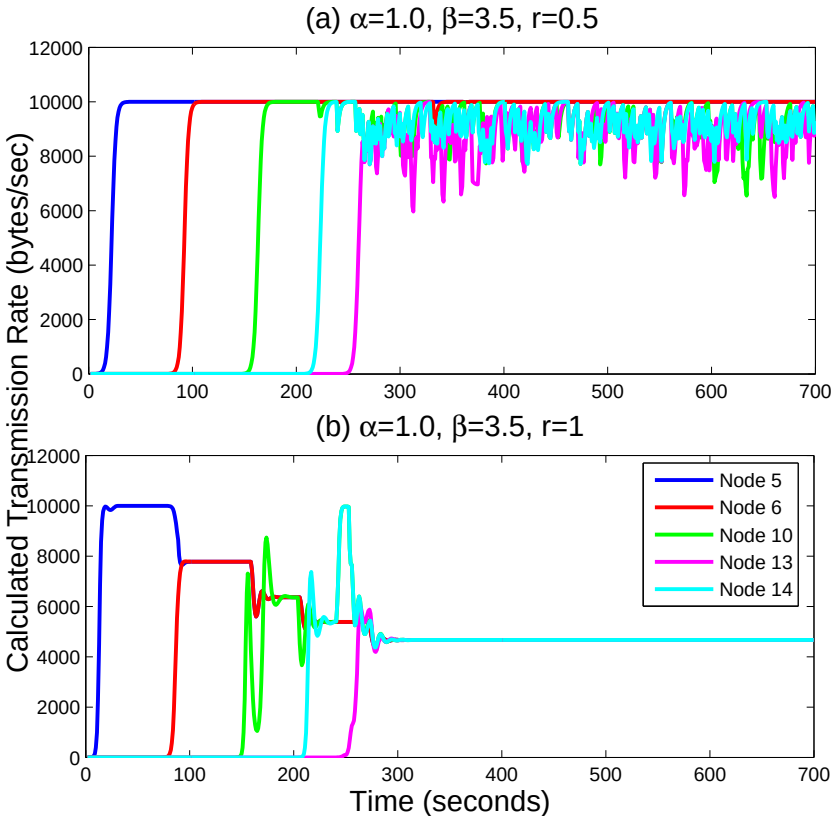


Figure 49: Calculated transmission rates for 5 active nodes scenario when $r = 0.5$ and $r = 1$.

that the results presented in this table consider only the scenarios where both the stability condition and the buffer overflow avoidance condition were satisfied.

Table 6 shows that in all scenarios, α values were significantly lower than β values. The values of α, that achieved the highest PDR in each scenario, ranged from 1.6 to 2.1, while the values of β ranged from 3.3 to 7.0. Results verified that as the number of active nodes scaled up, stable response of traffic flows and high PDR were achieved with the increase of parameter β (i.e. with the decrease of traffic flow transmission rates).

Table 6: Performance evaluations for realistic network conditions using NS2 [14].

α	β	*No. of Active Nodes*	*Packet Delivery Ratio*	
			1 Mbps	2 Mbps
1.6	3.3	3	0.981	0.999
1.6	4.3	5	0.993	0.999
1.9	6.5	7	0.961	0.986
2.1	7.0	10	0.892	0.951

In addition, the PDR decreased slightly with the increase in the number of active nodes. The decrease of PDR was attributed to the inadequacy of network resources (e.g. wireless channel capacity) to accommodate the traffic load injected from a large number of active nodes. When the MAC transmission rate increased to 2 Mbps, higher PDR values were observed (last column of Table 6) as a result of the enhanced channel capacity.

5.2.5 Sensitivity of Parameters

As shown above, it is beyond any doubt that the values of parameters α, β and r should be chosen carefully to ensure stability of traffic flows as well as buffer overflow avoidance. Results showed that parameters α and β are very sensitive to the number of active nodes within the sensor network. The understanding of how the LVCC model behaves in response to changes in the number of active nodes (or other network aspects e.g. topology, wireless channel conditions etc.) is of fundamental importance to ensure a correct and generalized use of this model. Sensitivity analysis can be used to order by importance the strength and relevance of the each varying factor in determining the variation in the output [93]. In models involving many

input variables sensitivity analysis is an essential ingredient of model building and quality assurance. Although sensitivity analysis is left for future work, this section gives some guidelines on setting parameters r, α and β according to the number of active nodes, based on the simulation results.

The parameter r can be set equal to 1 in order to preserve convergence to equilibria as well as smooth flow rate regulation, and simulation results do not show any sensitivity to varying factors. However, according to simulation results, the values of parameters α and β should be adapted by each sending node according to the total number of active nodes in the network as follows:

$$\alpha = \begin{cases} 1.6, & 1 \leq n \leq 5; \\ 2.1, & 6 \leq n \leq 10. \end{cases} \tag{24}$$

$$\beta = \begin{cases} 4.3, & 1 \leq n \leq 5; \\ 7.0, & 6 \leq n \leq 10. \end{cases} \tag{25}$$

In a realistic WSN, the sink node is aware of the total number of active nodes within the network. The sink node can piggyback this number on control packets that are periodically broadcasted within the network. Each node can further spread this information out over the network by means of control packets. Further study to allow for the adaptive setting of these parameters is encouraged for future work, which leads towards a real implementation.

5.2.6 Scalability and Fairness

Taking into consideration all the results presented thus far, the system proved to be adaptable against changing traffic load and achieved limited scalability by

sharing buffer capacity of nodes to their active upstream nodes. For example in Fig. 42, in the presence of one sender (node 5) the stable equilibrium point of the system given by Eq. 11 was 8.75 Kbytes/T (clusterhead node 1 transmitted at the same rate). When node 6 became active, each sender obtained 7 Kbytes/T, while the downstream node 1 (clusterhead) was able to accommodate both senders by increasing its rate using Eq. 20. When the number of senders scaled up, all senders could be supported by the system by diminishing the sending rate per node, thus offering graceful degradation. Fairness was also achieved having the available buffer capacity of each node equally shared among all activated flows. However, it must be stressed that as the number of nodes in the network increases, the scalability of LVCC is stressed. A realistic number of nodes is 20 with around the half of them being active at any point in time.

5.2.7 Comparative Evaluations

The proposed LVCC approach was compared with the traditional AIMD rate adaptation mechanism. The values of α, β and r were set to $2.4, 7$ and 1 respectively, while scenarios 1 and 3 were considered, involving 3 and 5 active nodes respectively, starting at different times as shown in the figures.

As shown in Figs. 50(a) and (b), the proposed LVCC approach achieved smooth throughput for each active node while maintaining friendliness among competing

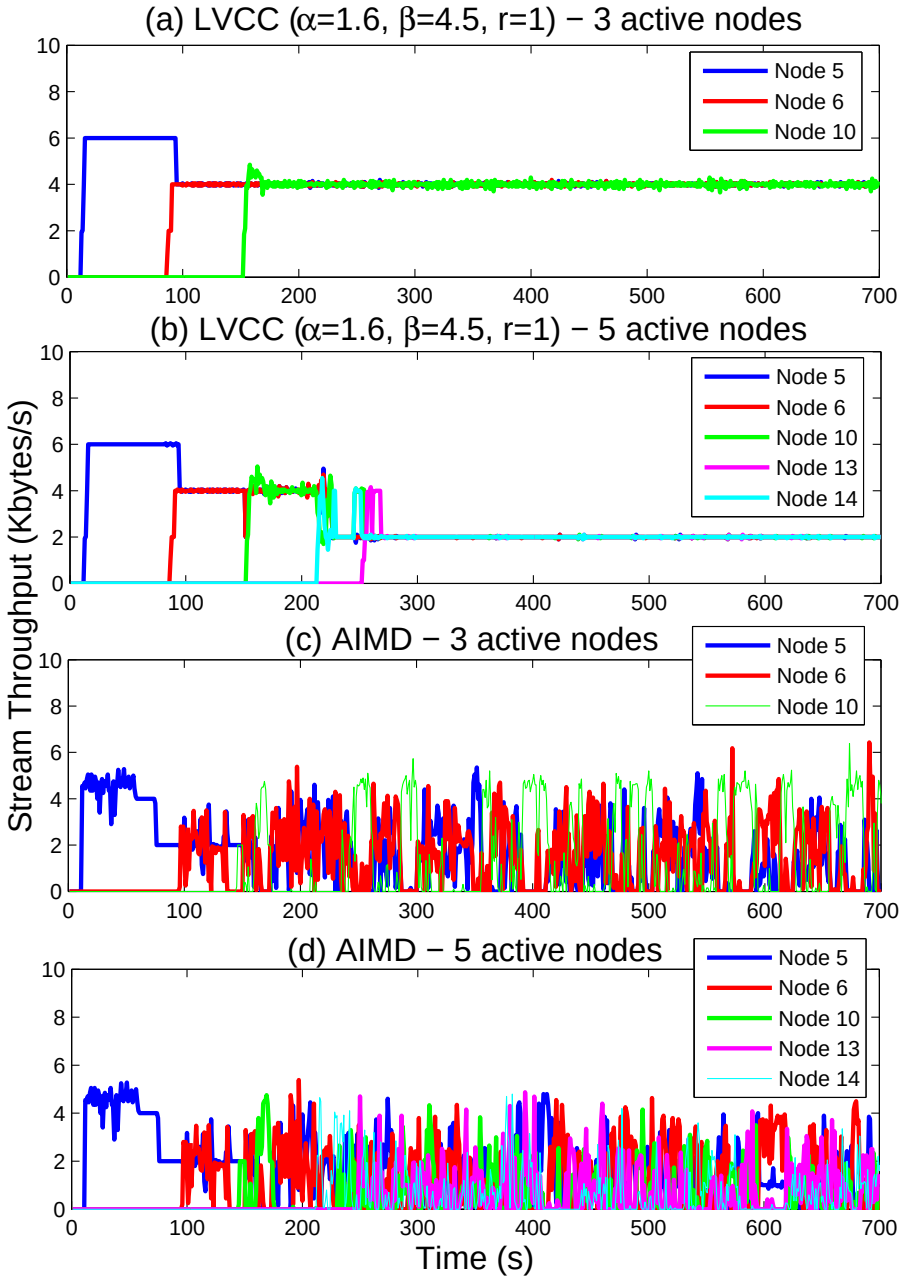


Figure 50: Throughput comparative evaluations between LVCC and AIMD for 3 and 5 active nodes.

flows. This controlled behavior is attributed to the LV-based calculated transmission rate evaluation which effectively and efficiently perceives the available network bandwidth, and fairly shares it among active nodes.

On the other hand, as can be seen in Figs. 50(c) and (d), the AIMD approach displayed a saw-tooth behavior which represents the probe for available bandwidth. The oscillations shown in Figs. 50(c) and (d) were attributed to multiplicative rate decrease after packet loss events. Therefore, the AIMD rate control policy seems to be ineffective in wireless environments due to the frequent occurrence of packet loss events. In addition, AIMD seems inefficient for streaming applications since the saw-tooth rate behavior may violate the QoS requirements of a stream and can lead to significant variation in streaming media quality. Furthermore, the end-to-end nature of the AIMD mechanism makes it incapable of operating in error-prone wireless multi-hop networks and results in reduced responsiveness, increased latency and high error rates, especially during long periods of congestion. Contrarily, the LVCC approach operates in a hop-by-hop manner providing fast responsiveness to changing network conditions.

5.3 Concluding remarks

This chapter evaluated the performance of both nature-inspired approaches under numerous scenarios of network operation. Both approaches proposed were effective (exhibiting robustness, self-adaptiveness, scalability) and efficient (simple to implement) solutions for avoiding congestion either by controlling the direction

of moving packets towards the sink (Flock-CC), or by controlling the amount of packets injected into the network (LVCC) under different topologies and changing network and traffic conditions. However, LVCC scalability is maintained efficiently for small scale networks (up to 20 nodes) with up to a certain number of active flows (around 10). Further increase of nodes or flows stresses scalability and may lead to instability.

Chapter 6

Conclusions

In this thesis, the problem of congestion in WSNs is addressed using techniques from nature. The aim of this study is to prevent or minimize congestion in WSNs by drawing inspiration from swarm intelligence and mathematical models of population biology.

Two novel nature-inspired congestion control approaches were proposed namely the flock-based congestion control (Flock-CC) approach and the Lotka-Volterra congestion control (LVCC) approach. We have first given an introduction to the problem of congestion in the field of wireless sensor networks as well as a detailed description of the two main pillars of this thesis: (a) the flocking behavior of birds and (b) the Lotka-Volterra competition model. Then, we have described both proposed congestion avoidance approaches, Flock-CC and LVCC, and next, we have presented a range of simulation tests in which we investigated the behavior and performance of these approaches. The tests were carried out in network simulator NS2. In what

follows, we first give an overview of the contributions and findings of this thesis, then discuss possible future research directions, and finally give a closing statement.

6.1 Contributions and findings of this thesis

Flock-CC mimics the synchronized group behavior of birds flocks and their ability to avoid obstacles in order to control the motion of packet flocks through a network of constrained sensor nodes. From the perspective of the emergent flocking behavior, the basic aim was to guide packets towards the sink whilst avoiding congestion regions. From the viewpoint of performance controlled WSNs, Flock-CC was aimed at providing high packet delivery ratio, low end-to-end delay and minimal energy tax to applications running on top of these networks, in a robust way. Flock-CC primarily targets real-time, densely-deployed, event-based wireless sensor networks.

The Flock-CC approach was influenced by the bio-swarm model of Couzin [11]. However, the Flock-CC approach differs in two aspects from Couzin's model: (1) The bio-swarm model of Couzin was formulated on the metrical (continuous three-dimensional) space, whereas the Flock-CC model is applied on a two dimensional topological (discrete) space defined by the graph of nodes, and (2) in Couzin's model (as well as in the Reynolds' model) individuals form flocks and move constantly in a given finite space without any attraction to a global target (final destination). On the other hand, in the Flock-CC approach, packets are expected to form flocks and move towards a global attractor (sink). The latter Flock-CC characteristic necessitates the existence of a field of attraction towards the sink.

In Flock-CC, each packet perceives repulsive and attractive forces exhibited by other packets located on neighboring nodes within the field of view, and decides a new hosting node on the basis of a desirability function. The desirability of each potential next hop node is adjusted to account for biased selection of nodes located closer to the sink. Also, perturbation, which allows exploration, was introduced to allow packets to pick random routes, and thus to avoid over-flooding popular (due to their low congestion levels) next hop nodes. The behavioral tendencies involved in Flock-CC were simulated to study its effectiveness in mimicking the collective behavior of bird flocks. Performance evaluations showed that Flock-CC was able to:

- alleviate congestion by balancing the offered load through alternative paths to the sink,
- offer acceptable PDR, above other competing approaches especially in high load scenarios, fast delivery (small EEDs) of packets to the sink and low energy tax,
- achieve adaptation to changing network and traffic conditions, robustness against failing nodes, even in extreme situations, scalability to different network sizes,
- outperform typical conventional congestion-aware and congestion-free routing approaches in terms of PDR in low, medium and high loads.

LVCC is a rate-based, hop-by-hop congestion control mechanism for small-scale multimedia streaming applications in WSNs. LVCC was designed on the basis of

the well known Lotka-Volterra competition model and is aimed at controlling the traffic flow rate at each sending node. In the context of LVCC, our contributions are summarized below:

- LVCC preserves the global properties of biological processes such as stability, self-adaptation, scalability and fairness, that are achieved collectively without explicitly programming them into individual nodes. Analytical evaluations and simulations were performed to understand how the variations of the model's parameters influence stability, sensitivity to parameters, scalability and fairness.

- Control system type simulations in Matlab validated the correctness of analytical results for plausible scenarios that could not be formally tested.

- Simulations showed that the proposed model achieves stability and smooth network operation under the analytically proposed conditions. Realistic scenarios of network operation and conditions were also simulated for effective parameter setting.

- Realistic scenarios evaluation suggested certain values for parameters α, β and r that are able to achieve high packet delivery ratio, low end-to-end delay, scalability and fairness among competing flows.

- LVCC was found to outperform AIMD-like rate-based congestion control approaches for WSNs in terms of stability and flow rate smoothness.

6.2 Future Research Directions

Here we point out some future research directions that are relevant for the work presented in this thesis. These concern further improvements on both Flock-CC and LVCC for coping effectively with additional network entities (e.g. multiple sinks, mobile nodes/sinks), the support of further QoS issues (e.g. multi-class priorities), and finally the use of the ideas behind Flock-CC and LVCC in other types of man-made systems.

6.2.1 Flock-CC

Interesting future work for the Flock-CC approach, is to investigate alternative methods of evaluating the attraction/repulsion forces and the desirability function. For example, alternative methods for establishing the artificial magnetic field and the attraction forces towards the pole(s) can be developed. In particular, beyond the hop distance, a new metric or technique can be devised for sink direction discovery and measuring the attractiveness to the sink. Also, energy reserves of sensor nodes can be taken into account when evaluating the desirability function.

Another interesting direction for future research, is to elaborate on the applicability of Flock-CC in the presence of *multiple sinks* can be investigated. In this study only one sink was considered in every scenario. However, in large-scale networks with a large number of sensor nodes, multiple sink nodes are expected to be deployed, not only to increase the manageability of the network, but also to reduce the energy dissipation at each node (packets will not need to 'travel' long

distances to reach the sink). In this context, the presence of multiple magnetic poles is expected to introduce new considerations during the evaluation of the desirability function and quite possibly will raise a need to devise criteria for differentiating the influence of each pole.

Another area of future work is the applicability of Flock-CC in the presence of one or more *mobile sinks*. In this study the sink node was always located in a fixed position. Results confirmed that energy consumption among sensor nodes was not uniform. In fact, the nearer a sensor node lies with relation to the sink node, the faster its energy was depleted. Also, in case of sensor node failure or malfunctioning around a sink node, the network connectivity and coverage may not be guaranteed. Therefore, mobile sink(s) can be used for balancing energy consumption among nodes and for finding alternative routes from source nodes to the sink(s). Also, multiple mobile sinks can be used to alleviate congestion as, for example, if they are 'guided' to move close to the area of an event. However, moving strategies for mobile sinks need to be developed.

Interesting future work would also be to apply the ideas behind Flock-CC in different kinds of man-made systems. One idea could be to use the set of rules at the heart of the Flock-CC approach to capture interactions in an urban road transportation system as for example vehicle-to-environment and vehicle-to-vehicle. Flock-CC rules can be used for navigating vehicles through congested road networks without any need for fixed infrastructure or centralized servers. In this context, cars moving towards a given city can be seen as individuals within a flock and the goal would

be to 'guide' cars form flocks and flow towards the global attractor (city) whilst trying to avoid congestion regions (congested roads, traffic lights etc). Toward this direction, quite recently, Google has developed technology for driverless cars [94]. The project team has a test fleet of at least eight, highly capable, vehicles equipped with numerous sensors in order to gather information from their environment. This information is processed by the car and in conjunction with stored or internet-based information are used to control the car's speed and to maintain its distance from other vehicles. Other types of systems in which Flock-CC's mechanisms could be useful are swarms of robots as well as unmanned aerial vehicle (UAVs) moving towards a given target. In such systems, individuals (e.g. robots, vehicles) operate in a dynamically changing environment with different kinds of obstacles and are in need of cooperating with each other in order to 'survive' and achieve their goals.

6.2.2 LVCC

For future work, it is planned to investigate if and under what conditions parameter values can be analytically optimized using conventional techniques, or even adopt other nature-inspired optimization techniques, such as genetic algorithms to adapt the tuning parameters on a global scale, over a much longer time period. Furthermore, due to the application-dependent nature of WSNs, wireless sensor networks deployed for different applications may require different congestion control approaches. In addition to the challenges for reliable data transport in WSN (e.g. packet loss, delay), there exist additional challenges due to the unique requirements

of the multimedia traffic, such as bounded delay and delay variation as well as minimum bandwidth demand. Therefore, a possible area of future work is to modify the LVCC approach in order to cope with a set of different priority classes (e.g. by means of unequal traffic rates) corresponding to different kind of traffic flows (i.e. different species in nature). These priority classes can be treated in a differentiated way by the congestion control algorithm.

The self-adaptive traffic flow regulation mechanism at the heart of the LVCC approach can be extended in the direction of transportation engineering. More specifically, this mechanism can be involved in controlling traffic flow injection into freeways/highways, i.e. to manage traffic flows on access ramps to freeways in order to avoid congestion phenomena, and thus delay for motorists. In this way, an autonomous real-time traffic injection control system could be able to minimize the overall delay for motorists according to the traffic input load and freeway congestion situation.

6.3 Closing statement

As shown in this thesis, nature-inspired techniques can be effectively adopted to handle communication networking problems that necessitate autonomous and decentralized operation. It is not only communication systems but also other man-made systems that are getting too complex for human operators to manage, and thus they should be able to operate autonomously. This means that systems should be self-organizing and self-adapting to dynamically changing conditions as well as

self-optimizing, self-protecting and self-healing in the presence of misbehavior, malfunction or failure. Nature serves as an excellent basis for developing autonomously evolving systems since the aforementioned self-* properties are intrinsic characteristics of natural systems.

Bibliography

[1] P. Antoniou, A. Pitsillides, T. Blackwell, and A. Engelbrecht, "Employing the flocking behavior of birds for controlling congestion in autonomous decentralized networks," in *2009 IEEE Congress on Evolutionary Computation*, A. Tyrrell, Ed., IEEE Computational Intelligence Society. Trondheim, Norway: IEEE Press, 18-21 May 2009.

[2] P. Antoniou, A. Pitsillides, A. P. Engelbrecht, T. Blackwell, and L. Michael, "Congestion control in wireless sensor networks based on the bird flocking behavior," in *4th IFIP TC 6 International Workshop on Self-Organizing Systems IWSOS*, ser. Lecture Notes in Computer Science, T. Spyropoulos and K. A. Hummel, Eds., vol. 5918. Springer, December 2009, pp. 220–225.

[3] P. Antoniou, A. Pitsillides, A. P. Engelbrecht, and T. Blackwell, "Mimicking the bird flocking behavior for controlling congestion in sensor networks," *Invited paper, 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, November 2010.

[4] P. Antoniou, A. Pitsillides, A. P. Engelbrecht, and T. Blackwell, "Applying swarm intelligence to a novel congestion control approach for wireless sensor networks," *Invited paper, 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, October 2011.

[5] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "Congestion control in wireless sensor networks based on bird flocking behavior," *submitted to the Elsevier Computer Networks Journal*, 2012.

[6] P. Antoniou and A. Pitsillides, "Towards a scalable and self-adaptable congestion control approach for autonomous decentralized networks," in *3rd European Symposium on Nature-inspired Smart Information Systems (NiSIS2007)*, St. Julians, Malta, November 2007.

[7] P. Antoniou and A. Pitsillides, "Congestion control in autonomous decentralized networks based on the lotka-volterra competition model," in *19th International Conference on Artificial Neural Networks ICANN, Part II*, ser. Lecture Notes in Computer Science, C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinas, Eds., vol. 5769. Limassol, Cyprus: Springer, September 2009, pp. 986–996.

[8] P. Antoniou and A. Pitsillides, "Congestion control in wireless sensor networks based on the lotka volterra competition model," in *Biologically Inspired Networking and Sensing: Algorithms and Architectures*, D. C. Verma and P. Lio, Eds. IGI Book, August 2010, pp. 158–181.

[9] P. Antoniou and A. Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the lotka-volterra competition model," *Elsevier Computer Communications, Special Issue on Applied Sciences in Communication Technologies*, vol. 33, no. 17, pp. 2039–2047, November 2010.

[10] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1987, pp. 25–34.

[11] I. D. Couzin, J. E. N. S. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, September 2002.

[12] G. D. Caro, F. Ducatelle, and L. M. Gambardella, "Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.

[13] L. Cobo, A. Quintero, and S. Pierre, "Ant-based routing for wireless multimedia sensor networks using multiple qos metrics," *Computer Networks*, vol. 54, pp. 2991–3010, December 2010.

[14] The Network Simulator NS-2, *http://www.isi.edu/nsnam/ns*. http://www.isi.edu/nsnam/ns/, 1995.

[15] W. Wiltschko and R. Wiltschko, "Magnetic orientation in birds," *The Journal of Experimental Biology*, vol. 199, pp. 29–38, 1996.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 1999.

[17] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[18] C. Sergiou, P. Antoniou, and V. Vassiliou, "Congestion control protocols in wireless sensor networks: A survey," *submitted to the IEEE Communications Surveys and Tutorials (accepted, subject to minor revision)*, 2012.

[19] B. Srean, J. Silva, L. Wolf, R. Eiras, T. Voigt, U. Roedig, V. Vassiliou, and G. Hackenbroich, "Performance control in wireless sensor networks: the ginseng project - [global communications news letter]," *Communications Magazine, IEEE*, vol. 47, no. 8, pp. 1 –4, august 2009.

[20] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, pp. 1–22, 2009.

[21] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2003, pp. 266–279.

[22] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM,

2004, pp. 134–147.

[23] Y. Sankarasubramaniam, O. Akan, and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in wireless sensor networks," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2003, pp. 177–188. [Online]. Available: http://dx.doi.org/10.1145/778415.778437

[24] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *Proceedings of the ACM SIGCOMM 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Pisa, Italy, September 11-15, 2006*, L. Rizzo, T. E. Anderson, and N. McKeown, Eds. ACM, 2006, pp. 63–74.

[25] T. Blackwell, "Swarming and music," in *Evolutionary Computer Music*, E. R. Miranda and J. A. Biles, Eds. Springer-Verlag, 2007, pp. 194–217.

[26] T. D. Burton, "Batman returns," USA/UK, 1992.

[27] R. Allers and R. D. Minkoff, "The lion king," USA, 1994.

[28] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, 1999.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948, November 1995.

[30] P. Gurfil, "Evaluating uav flock mission performance using dudek's taxonomy," *Proceedings of the 2005 American Control Conference*, vol. 7, pp. 4679–4684, 2005.

[31] G. Labonte, "Canadian artic sovereignty: Local intervention by flocking uavs," *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp. 1–8, 2009.

[32] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence.* NJ: John Wiley & Sons, Jan. 2006.

[33] A. J. Lotka, *Elements of Physical Biology.* Baltimore, MD: Williams and Wilkins, 1925.

[34] V. Volterra, "Variations and fluctuations of the numbers of individuals in animal species living together," *translation in R. Chapman, Animal Ecology*, pp. 409–448, 1931.

[35] R. Goodwin, "A growth cycle," in *Socialism, Capitalism and Economic Growth*, C. Feinstein, Ed. Cambridge University Press, 1967.

[36] V. Gaiko, "A generalized lotka-volterra model and its applications in epidemiology and immunology," in *The 6th International Conference: Dynamical Systems and Applications-2010*, July 2010.

[37] G. Hasegawa and M. Murata, "Tcp symbiosis: congestion control mechanisms of tcp based on lotka-volterra competition model," in *Interperf '06: Proceedings*

from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications sytems, vol. 200, no. 11. New York, NY, USA: ACM, 2006.

[38] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "From bird flocks to wireless sensor networks: A congestion control approach," Department of Computer Science, University of Cyprus, Tech. Rep. TR-11-5, September 2011. [Online]. Available: http://www.cs.ucy.ac.cy/ csp5pa1/publications/TR-11-5.pdf

[39] F. Brauer and C. Castillo-Chavez, *Mathematical Models in Population Biology and Epidemiology*. New York, NY, USA: Springer-Verlag, 2000.

[40] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, ser. Advanced Texts in Communications and Networking. John Wiley & Sons, 2010.

[41] M. Vuran, V. Gungor, and O. Akan, "On the interdependence of congestion and contention in wireless sensor networks," in *ICST SenMetrics, San Diego, CA*, July 2005.

[42] T. He, F. Ren, C. Lin, and S. Das, "Alleviating congestion using traffic-aware dynamic routing in wireless sensor networks," in *SECON*. IEEE, 2008, pp. 233–241.

[43] S. Chen and N. Yang, "Congestion avoidance based on lightweight buffer management in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 934–946, 2006.

[44] C. Sergiou, V. Vassiliou, and A. Pitsillides, "Reliable data transmission in event-based sensor networks during overload situation," in *WICON '07: Proceedings of the 3rd international conference on wireless internet*, Austin, Texas, October 2007, pp. 1–8.

[45] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum, "Reducing congestion effects in wireless networks by multipath routing," in *ICNP*. IEEE Computer Society, 2006, pp. 96–105.

[46] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multi-radio virtual sinks in sensor networks," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2005, pp. 116–129.

[47] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 148–161.

[48] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 786–795, 2007.

[49] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu, "Priority-based congestion control in wireless sensor networks," in *SUTC '06: Proceedings of the*

IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), vol. 1. Washington, DC, USA: IEEE Computer Society, 2006, pp. 22–31.

[50] B. L. C. Wang, K. Sohraby, "SenTCP: a hop-by-hop congestion control protocol for wireless sensor networks," *IEEE INFOCOM (Poster Paper)*, March 2005.

[51] Z. Xiangquan, G. Lijia, G. Wei, and L. Renting, "A cross-layer design and ant-colony optimization based load-balancing routing protocol for ad-hoc networks," *Frontiers of Electrical and Electronic Engineering in China*, vol. 2, no. 2, pp. 219–229, April 2007.

[52] J. C. Tillett, R. M. Rao, F. Sahin, and T. M. Rao, "Particle swarm optimization for the clustering of wireless sensors," *SPIE*, vol. 5100, no. 1, pp. 73–83, 2003.

[53] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, "Distributed particle swarm optimization and simulated annealing for energy-efficient coverage in wireless sensor networks," *Sensors*, vol. 7, no. 5, pp. 628–648, 2007.

[54] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 2000, pp. 39–42, 2000.

[55] M. Millonas, "Swarms, phase transitions, and collective intelligence," in *Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity*, C. G. Langton, Ed., vol. XVII. Addison-Wesley, 1994, pp. 417–445.

[56] Y. Liu and K. M. Passino, "Swarm intelligence: Literature overview," 2000.

[57] E. Shaw, "Schooling in fishes: Critique and review," *Development and Evolution of Behavior*, pp. 452–480, 1970.

[58] B. L. Partridge, "The structure and function of fish schools," *Scientific American*, vol. 246, no. 6, pp. 114–123, June 1982.

[59] S.-H. Lee, H. Pak, and T.-S. Chon, "Dynamics of prey-flock escaping behavior in response to predator's attack," *Journal of Theoretical Biology*, vol. 240, no. 2, pp. 250–259, 2006.

[60] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence (The Morgan Kaufmann Series in Artificial Intelligence)*, 1st ed. Morgan Kaufmann, April 2001.

[61] J. Krause and G. D. Ruxton, *Living in Groups*. Oxford: Oxford University Press, 2002.

[62] B. L. Partridge and T. J. Pitcher, "The sensory basis of fish schools: relative role of lateral line and vision," *J. Comparative Physiol.*, vol. 135, pp. 315–325, 1980.

[63] B. L. Partridge, "The structure and function of fish schools," *Sci. Am.*, vol. 245, pp. 90–99, 1982.

[64] W. Wiltschko, "Uber den einfluss statischer magnetfelder auf die zugorientierung der rotkehlchen," *Z. Tierpsychol.*, vol. 25, pp. 537–558, 1968.

[65] A. Von Middendorf, "Die isepiptesen rublands," *Mem. Acad. Sci. St Petersbourg VI*, vol. 8, pp. 1–143, 1859.

[66] C. Viguier, "Le sens de l"orientation et ses organes chez les animaux et chez l'homme," *Rev. phil. France Etranger*, vol. 14, pp. 1–36, 1882.

[67] D. Heyers, M. Manns, H. Luksch, O. Gunturkun, and H. Mouritsen, "A visual pathway links brain structures active during magnetic compass orientation in migratory birds," *PLoS One*, vol. 2, no. 9, p. e937, 2007.

[68] T. Yasuhiro and A. Norihiko, "The existence of globally stable equilibria of ecosystems of the generalized volterra type," *Journal of Mathematical Biology*, pp. 401–415, 1980.

[69] A. B. Roy and F. Solimano, "Global stability and oscillations in classical lotka-volterra loops," *Journal of Mathematical Biology*, vol. 24, no. 6, pp. 603–616, 1987.

[70] K. Murty, *Linear Complementarity, Linear and Nonlinear Programming*. Berlin, Germany: Heldermann, 1988.

[71] S. P. Dirkse and M. C. Ferris, "The path solver: A non-monotone stabilization scheme for mixed complementarity problems," *Optimization Methods and Software*, vol. 5, pp. 123–156, 1995.

[72] R. Kumar, H. Rowaihy, G. Cao, F. Anjum, A. Yener, and T. L. Porta, "Congestion aware routing in sensor networks," Department of Computer Science and Enginnering, Pennsylvania State University, Technical Report 0036, 2006.

[73] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, "Cluster-based congestion control for supporting multiple classes of traffic in sensor networks," in *EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 107–114.

[74] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theor. Comput. Sci.*, vol. 344, no. 2-3, pp. 243–278, 2005.

[75] R. Singh, D. K. Singh, and L. Kumar, "Swarm intelligence based approach for routing in mobile ad hoc networks," *International Journal of Science and Technology Education Research*, vol. 1, no. 7, pp. 147–153, 2010.

[76] S. Rajagopalan and C.-C. Shen, "Ansi: a swarm intelligence-based unicast routing protocol for hybrid ad hoc networks," *J. Syst. Archit.*, vol. 52, pp. 485–504, August 2006.

[77] G. D. Caro, F. Ducatelle, and L. M. Gambardella, "Swarm intelligence for routing in mobile ad hoc networks," *Swarm Intelligence Symposium*, pp. 76–83, June 2005.

[78] C. E. Perkins, "Ad hoc on demand distance vector (aodv) routing," *RFC 3561*, 1997.

[79] T. Iguchi, G. Hasegawa, and M. Murata, "A new congestion control mechanism of TCP with inline network measurement," in *Information Networking, Convergence in Broadband and Mobile Networking, International Conference,*

ICOIN 2005, Jeju Island, Korea, January 31- February 2, 2005, Proceedings, ser. Lecture Notes in Computer Science, C. Kim, Ed., vol. 3391. Springer, 2005, pp. 109–121.

[80] M. Analoui and S. Jamali, "A conceptual framework for bio-inspired congestion control in communication networks," in *BIONETICS '06: Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems*, vol. 275, no. 38. New York, NY, USA: ACM, 2006, pp. 1–5.

[81] M. Analoui and S. J., "Bio-inspired congestion control: Conceptual framework, algorithm and discussion," in *Advances in Biologically Inspired Information Systems: Models, Methods, and Tools*, ser. Studies in Computational Intelligence, F. Dressler and I. Carreras, Eds., vol. 69. Springer, 2007, pp. 63–80.

[82] M. Analoui and S. Jamali, "Nature-inspired congestion control: Using a realistic predator-prey model," in *Bio-inspired Modeling of Cognitive Tasks, Second International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC, Proceedings, Part I*, ser. Lecture Notes in Computer Science, J. Mira and J. R. Álvarez, Eds., vol. 4527. Springer, 2007, pp. 416–426.

[83] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 39–44.

[84] A. A. Ali, F. Michaut, and F. Lepage, "End-to-end available bandwidth measurement tools : A comparative evaluation of performances," in *Proceedings of the 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements*, ser. IPS-MoMe 2006, June 2006, pp. 1–14.

[85] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Gradient broadcast: a robust data delivery protocol for large scale sensor networks," *Wireless Networks*, vol. 11, no. 3, pp. 285–298, May 2005.

[86] M. Marsti, "Directed flood-routing framework for wireless sensor networks," in *In Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag, 2004, pp. 99–114.

[87] V. Lenders, M. May, and B. Plattner, "Density-based vs. proximity-based anycast routing for mobile networks," in *In IEEE Infocom*, 2006.

[88] G. G. Chen, J. W. Branch, and B. K. Szymanski, "Self-selective routing for wireless ad hoc networks," *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, vol. 3, pp. 57–64, August 2005.

[89] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[90] S. Selvakennedy, S. Sinnappan, and Y. Shang, "A biologically-inspired clustering protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2786–2801, October 2007.

[91] IEEE-802.11-WG., "Draft supplement to standard for telecommunications and information exchange between systems - lan/man specific requirements part 11: Wireless medium access control (mac) and physical layer (phy) specifications: Specification for radio resource measurement, ieee 802.11k/d9.0," *USA: The Institute of Electrical and Electronics Engineers, Inc*, September 2007.

[92] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[93] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis. The Primer*. John Wiley & Sons, Ltd, 2008.

[94] Wikipedia, the free encyclopedia, *Google driverless car*. http://en.wikipedia.org/wiki/Google_driverless_car, May 2012.