Ατομική Διπλωματική Εργασία

IMPLEMENTATION AND EVALUATION OF THE BIOLOGICALLY – INSPIRED ANTHOCNET ROUTING PROTOCOL IN SENSOR NETWORK

Stefanos Georgiou

University of Cyprus

Computer Science Department

29 May 2013

Ατομική Διπλωματική Εργασία

IMPLEMENTATION AND EVALUATION OF THE BIOLOGICALLY – INSPIRED ANTHOCNET ROUTING PROTOCOL IN SENSOR NETWORK

Stefanos Georgiou

University of Cyprus

Computer Science Deparment

29 May 2013

University of Cyprus

Computer Science Deparment

IMPLEMENTATION AND EVALUATION OF THE BIOLOGICALLY – INSPIRED ANTHOCNET ROUTING PROTOCOL IN SENSOR NETWORK

Stefanos Georgiou

Supervisor Academic
Dr. Andreas Pitsillidis

This Thesis Project implement in purpose to acquire my bachelor degree
in Computer Science Department in University of Cyprus

29 May 2013

Acknowledgments

I would like to express my gratitude to my advisor Prof. Andreas Pitsillides for his continuous guidance, support, and patience during this research. It was honor to have him as my advisor. In addition, I would like to thank Dr. Pavlos Antoniou, for his valuable help and knowledge, insightful comments and being always present to answer all my questions and to guide me.

Last but not least, I would like to thank my family and all my friend for their support and significant help during one of the hardest periods in my student life. For all my friends who motivated me. Special thanks to Constandinos Fasoulis, Charis Maragkos, Nikolas Kleanthous, Andreas Lymbouras, Zacharias X'Lambrou and Xrusi Sea. During this period also some lectures who inspired me and motivated me to become better and more professional in my field of study. Special thanks to Dr. Dimitris Zeinalimour, Dr. Giannos Sazeidis and Dr. Kolios Panagiwtis.

Abstract

This work is related with the problem of routing and congestion in wireless sensor networks (WSNs). In particular, it implements and evaluates a robust and self-adapting nature-inspired congestion control protocol for real-time event-based applications. We study the problem of congestion in WSNs from the perspective of nature-inspired design. The advantage of this approach is that it simplifies the implementation of every node, and requires minimal information exchange, while maximizing network lifetime and providing graceful performance degradation. Our main source of inspiration is the field of Ant Colony Optimization (ACO). The adaptively and robustness of ACO algorithms allow handling the challenges presented in the environment. We present the AntHocNet routing algorithm, which combines ideas from ACO routing with techniques from dynamic programming and other mechanisms taken from more traditional routing algorithms. The main idea of the implementation of the AntHocNet is to 'guide' packets (ants) to flow towards the sink (global attractor), whilst trying to avoid congestion regions (obstacles). AntHocNet is an implementation of both elements reactive and proactive routing. Specifically, it combines a reactive route setup process with a proactive route maintenance and improvement process. After the creating of routes nodes start the diffusion process where they communicate with each other to check the link availability every interval of time. The communication is done between nodes which are connected to the routes to minimize the route maintains effect. If failure occurs they try to avoid it by finding new route for their packets or if the current route is heavily congested nodes try to find new neighbors.

We present a detailed description of our implementation of this algorithm within the NS-2 network simulator. We perform a thorough evaluation of our implementation using a number of scenarios under various network congestion conditions. We perform a guided exploration of the control parameter space to validate our implementation and reach an optimal combination for the tested scenarios. We also perform a detail examination of the different internal stages of the protocol for a better understanding of its inner working. And finally, we compare the performance of the protocol with the FlockCC protocol and show that for the scenarios and control parameters we used the AntHocNet may or may not achieves better performance.

Contents

Chapter 1

Introduction

1.1 General Introduction

The main purpose of this thesis was to continue and complete a previous thesis about AntHocNet with WSN thesis which started by Sandy Agias Mixail. During this project Sandy didn't have the time to complete the route failure repair process and also important aspects were missing from this project as nominated by G. Di Caro, Ducatelle and L.M. Gambardellas project [12] Generally in this thesis we investigated the development of adaptive routing algorithms for ad hoc wireless multi-hop networks using techniques from artificial intelligence, and in particular from swarm intelligence. The thesis has the following:

a) To check previous work and also correct and rewrite some parts of the missing algorithm implementation.
b) Evaluation of the AntHocNet in the context of WSN, because most previous works are related to Ad-hoc systems and not in WSN.
c) Evaluation of the implemented algorithm and comparison with other protocols/algorithms.

The main challenge facing protocol designers is to create a high reliability and low energy tax, transportation protocol with the following properties: (a) low number of collisions and retransmissions, (b) low packet loss resulting in high packet delivery ratio, (c) low latency, and (d) high fault tolerance. In sensor networks, packet bursts can be dynamically and randomly initiated at any sensor node within the network which is expected to arrive to the sink once the occurrence of a given event has been detected.[4]

One of the features of WSNs comparing to MANETs is that WSNs have no mobility. In this case the topology will not change rapidly and there is a limited need of route maintenance which means proactive process will not occur many times.

Recently, the increased use of wireless communication, is one of the most important developments in this area. Consequently, many different wireless technologies have grown explosively and made wireless communication networks one of the most important areas of research in computer science. In this thesis, I will focus on one important type of wireless networks: ad hoc wireless multi-hop networks (AHWMNs)[4]. AHWMNs are communication networks that consist of wireless nodes with an ad hoc displacement. The Nodes satisfy the following assumptions: they have routing capabilities, they can forward data packets towards other nodes and they can join and leave the network at any time[8].

WSNs, in many ways, can be likened to social groups (with nodes 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 the major strengths bio-systems as they must respond to addition or removal of members, as well as to sudden changes in the environment. In this work we utilize ideas adopted from the behavioral tendencies of natural system for designing robust network control techniques. Drawing inspiration from the collective behavior of 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 nature-inspired model[4].

In this thesis, we focus on the problem of routing in AHWMNs. Routing is the task of selecting and maintaining the paths in a network, and is responsible for connecting the remote source to the destination nodes. Routing is complicated in AHWMNs because they are:

a) Dynamic networks: The AHWMNs are dynamic because the connections between nodes in the network are set up in an unplanned manner, and are often changed while the network is in use so the routing algorithm should be adaptive.

b) The wireless communication is unreliable: Data and control packets can easily get lost during transmission, especially when multiple transmissions take place simultaneously and interfere with each other and these packets are not retransmitted. A routing algorithm should be robust.

c) Because WSNs are used to collected information from different areas and for long period of time (weeks, months, years) we need the routing algorithm to be efficient. The network bandwidth, node processing power, memory, battery power are limited: A routing algorithm should be efficient in case the network can "live" for longer period of time.

d) Usually the network size is huge: With the ever growing numbers of portable wireless devices, many AHWMNs are expected to grow to very large sizes. Routing algorithms should be scalable.

e) Also the main purpose of these routing protocol is to forward information to a sink node. By the forwarding and receiving capability of the nodes results will be extracted in case to compare with FlockCC algorithm[4].

To solve the challenging problem of routing in AHWMNs, we use techniques from swarm intelligence (SI)[8] and ant colony optimization (ACO)[8]. SI is the collective behavior of decentralized, self-organized systems, whose concept is from artificial intelligence that is focused on the design of algorithms inspired by the collective behavior of social insects and other animal societies. A subcategory of SI is ACO that takes its inspiration from the foraging behavior of ants living in colonies. Through the movement of the ant in the network, from source to destination nodes ants which are the control packet of these process leaves pheromone information in their way to make the routes more attractive to the data packets.

In particular, in this work we implement a protocol (AntHocNet[8]) that provides congestion avoidance by mimicking ants behavior, where packets are modeled as ants moving over a topological space, e.g. a sensor network. The main idea of the implementation of the AntHocNet is to 'guide' packets (ants) to flow towards the sink (global attractor), whilst trying to avoid congestion regions (obstacles). AntHocNet contains elements from both reactive and proactive routing. Specifically, it combines a reactive route setup process with a proactive route maintenance and improvement process. AntHocNet was in the first place inspired by the ACO approach to routing. This is evident in the way that it gathers, stores and uses routing information. Nevertheless, AntHocNet contains elements from distance vector routing. In particular, the information gathering process used in its proactive route maintenance and improvement process combines the route sampling strategy from ACO routing with an information bootstrapping process that is similar to the one used in distance vector routing algorithms. The way both approaches are combined is novel and allows the algorithm to get the best of both worlds[8].

In this thesis , I investigated how the ideas from ACO routing can be used to build an adaptive and efficient algorithm with robustness. My current work is to implement and evaluated ACO fundamentals using WSNs network. After these I through different parameters and compare them with FlockCC routing algorithm and make a conclusion[3,4].

1.2 Contribution of this thesis project

This thesis has the following contributions:

- implementation of AntHocNet algorithm and adapted it for Wireless Sensor Networks. We have slightly modified the original algorithms to improve its performance and also to correct some mistakes from the previous work.
- We present a thoroughly validation and evaluation of the implemented protocol and its properties when utilized in the context of WSN under various configuration and conditions and parameters.
- We present a quantitative comparison with another biologically inspired routing protocol including path discovery, sink packet delivered, packet lost , end-to-end delay and energy efficiency.

1.3 Outline of the thesis project

The rest of this thesis is organized as follows: In Chapter 2 we provide background information and previous work related to the topics discussed in this thesis. In Chapter 3 we present a detailed description of the AntHocNet algorithm and our implementation. A detailed evaluation of the implemented algorithm is presented in Chapter 4 and Chapter 5 concludes this thesis and discusses future work.

Chapter 2

Literature Review

2.1 Ad Hoc Wireless Multi-hop Networks

Wireless communication technologies is ever increasingly utilized by many of the devices in our current times. More and more of the electronic devices is requiring to “communicate” with each other using technologies like Bluetooth, WiFi, ZigBee etc. Wireless networks exist in many types and configurations. The emphasis of this thesis is on a specific type called Ad Hoc Wireless Multi-hop Networks but the evaluation of our algorithm will be implemented with WSNs(AHWMN)[8].

AHWMN is a network consisting of nodes that communicate exclusively through wireless connections, in which data can be forwarded over multiple hops. Such nodes are at least partly deployed in an *ad hoc* manner. Ad hoc deployment entails that little or no planning is needed, and so changes in the network (such as adding, moving or removing nodes) can be done with minimal extra work.[8]

Differences between AHWMNs and traditional telecommunication networks[8]:

a) The most important difference between AHWMNs and traditional telecommunication networks is that the topology of an AHWMN is dynamic. In this case the routing algorithm should be act proactively to maintain the routes or act reactively in case to find new routes because the topology is not stable. This means that it can be extended by adding new wireless nodes, reduced by removing nodes, or changed in a continuous way if some of the nodes are mobile. As such, no element of planning is involved, but rather the topology emerges from the dynamic placement of the nodes which are connected with each others over the wireless radio signal. Note that at any time, only a sub-part of the total nodes is “visible” by each node.

b) Another difference is that AHWMNs data’s transport is less reliable. Moreover there exist less available bandwidth because they rely on wireless links. And the nodes typically have limited resources so they energy requirements stays low to be active for longer time(memory, power, etc..).

c) Furthermore, it is difficult to optimize the use of network resources because AHWMNs are decentralized.

d) AHWMNs are expected to grow to very large sizes, and so scalability is important.

Difference between AHWMNs and WSNs:

a) In AHWMSs we have mobile nodes which cause the constant change of the nodes position and the topology of the network and that is the main reason we need to discover routes more often than using WSNs which are immobile nodes. That also means in WSNs we will have less needs for routes discovery and our network topology changes will caused only when nodes are failing or nodes have high congestion and we will need to find new routes for our data packets.

b) AHWMNS are consuming more energy rather than WSNs mobile nodes are changing position and we have more collisions and more packet will fail and we have more need for retransmissions. If no nodes are in range nodes cannot send and will stay idly.

AHWMNs are classified into:

a) Wireless mesh networks (WMNs) [13] are more heterogeneous than other AHWMN. They consist of mesh client nodes and mesh router nodes, that are less mobile but have more resources. Mesh clients are mobile, and usually communicate through one wireless interface (an antenna). They appear as end points of data traffic and as routers. Mesh routers are more static, and they are more powerful devices than the mesh clients. Moreover they support many different wireless technologies. Mesh routers can create a structured organization and can improve the applicability and the capacities of the network. Hence, the energy consumption of this process is high enough to maintain large number of routes and connections sending Hello packet to update node time tables.

b) Mobile ad hoc Networks (MANETs) [11] are networks that are made up of a set of homogeneous mobile devices. They use an antenna to communicate through wireless connections. In ad-hoc networks, nodes are not familiar with the topology of their networks. Instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbors. Each node learns about nodes nearby and how to reach them, and may announce that it, too, can reach them. [25] MANETs are dynamic, flat, and fully decentralized networks without central control or overview. MANET algorithms are typically highly adaptive to the ever changing environment, they are bound to be robust in order to deal with unreliable wireless transmissions, they should work in a fully distributed way and be efficient in their use of the limited network resources, such as bandwidth and power. All nodes have the capability to send/receive data and relay/route traffic because they are all homogeneous without a specialization. MANETs have different routing protocols categories as Table-driven , On Demand, Flow-oriented, Hybrid and Hierarchical routing protocols.

c) Sensor networks [1] are AHWMNs that consist of wireless sensor nodes (WSNs). WSNs have been 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 for extended periods of time. Typically, WSNs comprises a small (and often cheap) cooperative devices (nodes), which may be (severely) constrained in terms of computation capability, memory space, communication bandwidth and energy. That is one of the main need of the routing protocol has to be low on energy consumption and transfer data stochastically. WSNs can be used for acquiring different information like images from battlefields , temperatures, sounds, and many more information in different subjects.

In the context of WSNs, 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 sink nodes. In this thesis project the main purpose is to transfer as many information to a single node(sink) that represents the data collection of the project through different case scenarios. This mass of interactions, in conjunction with variable wireless network conditions, may result in unpredictable behavior in terms of traffic load variations and link capacity fluctuations also many events can occur in a network. The network condition is worsened due to topology changes driven by node failures, mobility, or intentional misbehavior or even with the high data packet rate which can cause congestion and collisions in the network. These stressful situations are likely to occur in WSN environments, thus increasing their susceptibility to congestion. Problems specific to sensor networks stem from the fact that sensor nodes are small and have very limited resources for storage, processing and transmission, so that highly efficient algorithms are needed which can be adaptive and robust. The main problem with energy reserves and basically their batteries which can not be replaced(single use devices). Moreover, the use of cheap, low power radio technology also means that communication is highly unreliable and irregular so algorithms have to be robust and should be able to deal with unidirectional links. Another issue in sensor networks is that their topology is usually very dynamic because sensor can fail often and the need of new routes discovery is occur, in this case means more energy usage is needed every time when nodes are failing and we don't have new routes for our data packets. Limited power and energy reserves is the main reason for failure events in WSNs. What exacerbate even more the situation is the fact that often new sensor devices are added since in such networks vast numbers of nodes are typically deployed. Finally, data traffic patterns show certain characteristic regularities[3]. Also high level of congestion is a reason why the need of discovering new routes is occur in purpose to lower the congestion level and collisions that may occur.

2.2 Wireless Sensor Networks

The unpredictable nature of WSNs necessitates robust, self-adaptive, and scalable mechanisms which are very basic to the mission of a successfully implementation of data transferring in random network topologies in the environment. The focal point of this study is to design a robust and self-adaptive congestion control (CC) mechanism for delivering enhanced application fidelity at the sink (in terms of packet delivery ratio and delay) under varying network conditions, inspired by nature. This implementation should be simple at individual node level with minimal exchange of information during different parameters and situations. [4]

The problem of congestion in WSNs, refers to the symptoms and the consequences of congestion, and presents a number of recent CC approaches. Congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. Typical effects include queuing delay, packet loss and blocking of new connections. Also a result of the queue delay and congestion in the network is the end-to-end delay of the data packet. A consequence of the latter two is that incremental increases in offer load lead either only to small increases in network throughput, or to an actual reduction in network throughput[3]. Network protocols which use aggressive retransmissions to compensate for packet loss tend to keep systems in a state of network congestion even after the initial load has been reduced to a level which would not normally have induced network congestion. Thus, networks using these protocols can exhibit two stable states under the same level of load. The stable state with low throughput is known as Congestive Collapse[23]. WSNs operate under large, sudden, and correlated-synchronized impulses of packets, that may suddenly arise in response to a detected or monitored event. All packets must be directed towards one or more sink nodes or pick new routes upon congestion events in case to avoid more retransmissions.

Types of congestion phenomena in WSNs:

1. Node-based: A node is accumulating packets in its buffer when its outgoing channel capacity is exceeded by a huge income of traffic. In our implementation we keep the router buffers typically low at 50 packets and we use scenarios data packet rate of 25, 35 and 45 packets per second. If the traffic is persistent then the buffer capacity of the node is not able to handle the long queues and as result it overflows occurring long delays and in the end when queue is full node is starting to drop tail packets and the need of retransmission occurs. That cause energy waste and time consumption in our network.

2. Link-based: Link- based congestion can be caused from the multi-hop nature of WSNs, the shared communication medium and the limited bandwidth.

   In wireless networks, local channel contention arises in the vicinity of a sensor node due to the limited bandwidth and interference among multiple neighboring nodes that try to access the wireless medium simultaneously. As a result, the time variant nature of the outgoing channel capacity makes the congestion level fluctuating and unpredictable.[3]

Congestion Consequences are energy waste, throughput reduction, increase in collisions and retransmissions at the MAC layer, increase of queuing delays and even information loss, leading to the deterioration of the offered quality of service (QoS), decrease of network lifetime and even the decomposition of network topology in multiple components[3].

2.3 Swarm Intelligence

Real network topologies are typically very large with thousands of nodes, so algorithms handling such sizes need to scale well. Swarm intelligence (SI)[24] is inspired from nature that deals and solve successfully and effectively with similar types of complex problems. Swarm intelligence is the collective behavior of decentralized, self organized natural or artificial systems. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking, animal herding, bacterial growth etc. The application of swarm principles to robots is called swarm robotics, while 'swarm intelligence' refers to the more general set of algorithms.[24]

SI techniques, motivated by the collective behavior of social insect societies living in decentralized, self-organizing, and adapting environments, reportedly provide a promising basis for computing environments that need to exhibit these characteristics and more specifically by the ants behavior and movement [17]. 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. 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 [10]. Social groups found in nature (e.g. ant colonies, bird flocks, etc.) carry out their tasks collectively in order to contribute to a common goal. Even though individuals coordinate to accomplish a given global mission in a complex world (e.g. foraging, migration, nest building, defend against predators, etc.), an individual has only local perception of the surrounding environment and exhibits specific behavioral tendencies which are governed by a few simple rules.[24]

2.4 Ant Colony Optimizations

Routing is the task of selecting and maintaining the paths in a network, and is responsible to connect the remote source to the destination nodes. During failure the need of reaction is occurred. As mentioned above, routing is complicated in AHWMNs because they are:

a) Dynamic networks: This is the main reason why in AHWMNs the need of maintaining routes and update periodically is high. Most times the connections between nodes and the network is occurred from unplanned event. That why its important that algorithm should be adaptively and robust due to the network changes.

b) The wireless communication is unreliable: Data and control packets can easily lost during transmission, especially when multiple transmissions take place simultaneously and interfere with each other. Especially when multiple nodes try to sent packet to the same node the collision phenomenon will also occur. A routing algorithm should be robust.

c) The network bandwidth, node processing power, memory, battery power are limited: A routing algorithm should be efficient in case to last longer period of time.(also because batteries of sensor devices cannot be replaced)

d) Usually the network size is huge: With the ever growing numbers of portable wireless devices, many AHWMNs are expected to grow to very large sizes. Routing algorithms should be scalable.

To solve the challenging problem of routing in AHWMNs, we use techniques from swarm intelligence (SI) and ant colony optimization (ACO)[8]. A subcategory of SI is ACO that takes its inspiration from the foraging behavior of ants living in colonies[8].

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. In the natural world, when ants find food, they return to their colony while laying down pheromone trails so if other ants find such a path, they follow the trail, returning and reinforcing it if they eventually find food. The pheromone trail starts to evaporate over time, so its attractive strength is reduced. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. [24]

For example a short path, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. When one ant finds a good path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants following a single path. These paths then attract more ants, which in turn increases their pheromone level, until there is a convergence of the majority of the ants onto the shortest paths. The ants completing paths can be seen as repetitive samples of possible paths, while the laying and following of pheromone results in a collective learning process guided by implicit reinforcement of good solutions. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around our network topology representing control packets.[22].

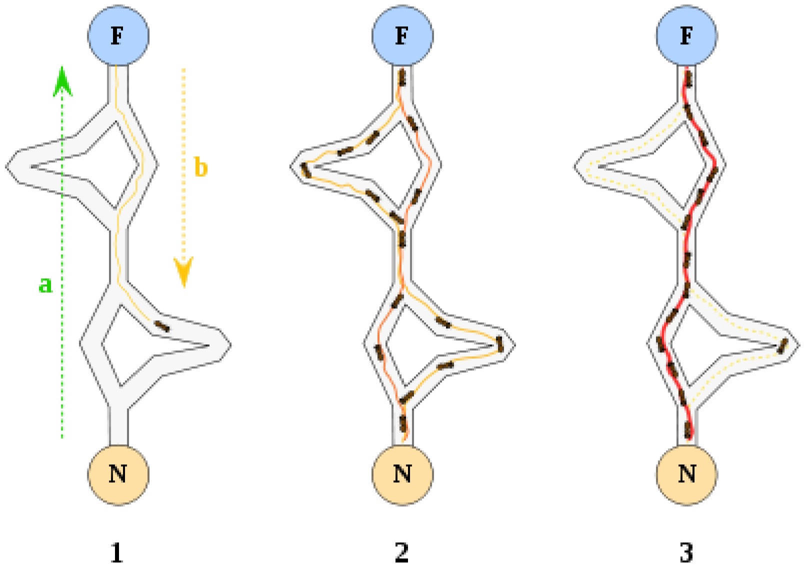


Figure1 shows an example of how ants find the shortest path between food source and the next[22].

1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b)
2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.
3. Ants take the shortest route, long portions of other ways lose their trail pheromones.

In a series of experiments[22] on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route. A model explaining this behavior is as follows:

1. An ant (called “blitz”) runs more or less at random around the colony;
2. If it discovers a food source, it returns more or less directly to the nest, leaving in its path a trail of pheromone;
3. These pheromones are attractive, nearby ants will be inclined to follow, more or less directly, the track;
4. Returning to the colony, these ants will strengthen the route;
5. If there are two routes to reach the same food source then, in a given amount of time, the shorter one will be traveled by more ants than the long route;

6. The short route will be increasingly enhanced, and therefore become more attractive;
7. The long route will eventually disappear because pheromones are volatile;
8. Eventually, all the ants have determined and therefore “chosen” the shortest route.

ACO algorithms for routing in networks differ from traditional algorithms. They gather routing information through the repetitive sampling of possible paths between source and destination nodes using artificial ant packets. Probabilistic distance vector tables, called pheromone tables, fulfill the role of pheromone in nature, with artificial ants being forwarded along them in a hop-by-hop way using stochastic forwarding decisions. Also data packets are forwarded stochastically using similar tables, resulting in automatic data load balancing and by following the highest pheromone places firstly by control packets[8].

As mentioned before the main source of inspiration behind ACO is a behavior that is displayed by ants finding the shortest path during foraging[17]. During the reactive ant process sources sent out in the network many ant (control packets) which their main purpose is to find the food(sink). After the discovery of the food ant goes back to sources nodes following the packet list and also leaving pheromones on the nodes . Sink nodes keeps only one route from each source node and keeps only the first ants form each nodes considering that first arrived ant have the shortest paths. Keeping the shortest paths is equal to low energy consumption of the network but also means hi chances for collisions and end-to-end delay. Pheromone is the most important aspect of this implementation. Pheromones are kept in each nodes pheromone table and its updated through time. Pheromones are used by many different species of social animals for a wide variety of tasks that involve coordinated behavior.

The shortest path finding process of the ants is highly distributed and self-organized. There is no central control mechanism neither infrastructure , the organization of the behavior emerges from the simple rules communications via pheromones. Second, it is highly robust and efficient. This is related to the property of self-organization: the system has no single point of failure, but instead consists of a high number of individually unimportant agents, so that even significant agent losses do not have a large impact on the performance. Third, the process is adaptive. Since none of the ant behavior is deterministic, and some individuals keep exploring also longer paths, the system can adapt to changes in the environment. Finally, the process is scalable: the process can be scaled to arbitrarily large colonies[8]

2.5 FlockCC Protocol

Flock Congestion Control protocol[3,4] implemented by Dr.Pavlos Antoniou and its another routing protocol that draws inspiration from biological mechanisms from species of social animals. In particular, inspiration is drawn from the flocking and obstacle avoidance behavior of birds to 'guide' packets bypass obstacles like congested regions and dead node zones in order to avoid the link failures and also on to avoid congestion. Recent studies showed that the flock-based congestion control (Flock-CC) approach is robust, self-adaptable and energy-efficient, involving minimal information exchange and computational burden when used in uniform grid topologies. Flock-CC demonstrated robustness against failing nodes, and outperformed other congestion-aware routing approaches in terms of packet delivery ratio, end-to-end delay and energy tax. It considers to have low end-to-end delay and hi delivery ration plus low energy tax and that is why we wanted to compare it with AntHocNet which is also adaptive, robust and energy efficient hybrid routing algorithm.

The proposed approach mimics the flock flocking behavior of birds, where packets are modeled as birds flying over a topological space (sensor network). The packets are generated by sensor nodes and are 'guided' to form flocks and 'fly' towards a global attractor (sink), whilst trying to avoid obstacles (congested regions). The direction of motion is influenced by (a) repulsion and attraction forces exercised by neighboring packets, as well as (b) the gravitational force in the direction of the sink. The flock-based congestion control (Flock-CC) approach provides congestion detection on the basis of node and channel loading and traffic redirection over multiple paths[3].

The proposed approach involves reference to artificial bird flocks consisting of individuals with finite range of view (perception) which interact with each other as well as with the environment. The behavior of each individual is influenced by other individuals within its neighborhood. Typically, four simple behavioral rules govern the individual behavior:

1) repel from neighbors (if too close) to avoid collisions,
2) attract to neighbors (if apart) to maintain coherence among the members of a flock,
3) match velocity (speed and direction) with neighbors, and
4) introduce a random element that allows for exploration.

The behavior and the feature of the FlockCC routing protocol makes it perfect for comparison between AntHocNet.

2.6 Basic Definitions from Networking

In this section we provide the unfamiliar reader with a brief definition of the technical terms we use often in this thesis as also provided by [24]. Part of the definitions has been adapted from[8].

The physical layer

In the seven-layer OSI model of computer networking, the physical layer or layer 1 is the first (lowest) layer.[1] The implementation of this layer is often termedPHY.

The physical layer consists of the basic networking hardware transmission technologies of a network.[2] It is a fundamental layer underlying the logical data structures of the higher level functions in a network. Due to the plethora of available hardware technologies with widely varying characteristics, this is perhaps the most complex layer in the OSI architecture.

The physical layer defines the means of transmitting raw bits rather than logical data packets over a physical link connecting network nodes. The bit stream may be grouped into code words or symbols and converted to a physical signal that is transmitted over a hardware transmission medium. The physical layer provides an electrical, mechanical, and procedural interface to the transmission medium. The shapes and properties of the electrical connectors, the frequencies to broadcast on, the modulation scheme to use and similar low-level parameters, are specified here.

Within the semantics of the OSI network architecture, the physical layer translates logical communications requests from the data link layer into hardware-specific operations to affect transmission or reception of electronic signals.[26]

The data link layer

In computer networking, the link layer is the lowest layer in the Internet Protocol Suite (commonly known as "TCP/IP"), the networking architecture of the Internet (RFC 1122, RFC 1123). It is the group of methods or protocols that only operate on a host's link. The link is the physical and logical network component used to interconnect hosts or nodes in the network and a link protocol is a suite of methods and standards that operate only between adjacent network nodes of a Local area network segment or a wide area network connection.

Despite the different semantics of layering in TCP/IP and OSI, the link layer is often described as a combination of the data link layer (layer 2) and the physical layer (layer 1) in the Open Systems Interconnection (OSI) protocol stack. However, TCP/IP's layers are descriptions of operating scopes (application, host-to-host, network, link) and not detailed prescriptions of operating procedures, data semantics, or networking technologies.

RFC 1122 exemplifies that local area network protocols such as Ethernet and IEEE 802, and framing protocols such as Point-to-Point Protocol (PPP) belong to the link layer.[27]

The transport layer

In computer networking, the transport layer or layer 4 provides end-to-end communication services for applications[1] within a layered architecture of network components and protocols. The transport layer provides convenient services such as connection-oriented data stream support, reliability, flow control, andmultiplexing. Transport layers are contained in both the TCP/IP model (RFC 1122),[2] which is the foundation of the Internet, and the Open Systems Interconnection (OSI) model of general networking. The definitions of the transport layer are slightly different in these two models. This article primarily refers to the TCP/IP model, in which TCP is largely for a convenient application programming interface to internet hosts, as opposed to the OSI-model definition of the transport layer.

The most well-known transport protocol is the Transmission Control Protocol (TCP). It lent its name to the title of the entire Internet Protocol Suite, TCP/IP. It is used for connection-oriented transmissions, whereas the connectionless User Datagram Protocol (UDP) is used for simpler messaging transmissions. TCP is the more complex protocol, due to its stateful design incorporating reliable transmission and data stream services. Other prominent protocols in this group are the Datagram Congestion Control Protocol (DCCP) and the Stream Control Transmission Protocol (SCTP).[29]

The Network Layer

In the seven-layer OSI model of computer networking, the network layer is layer 3. The network layer is responsible for packet forwarding including routingthrough intermediate routers, whereas the data link layer is responsible for media access control, flow control and error checking.

The network layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination host via one or more networks, while maintaining the quality of service functions.

Functions of the network layer include:

- Connection model: connectionless communication

  For example, IP is connectionless, in that a datagram can travel from a sender to a recipient without the recipient having to send an acknowledgement. Connection-oriented protocols exist at other, higher layers of the OSI model.

  - Host addressing

    Every host in the network must have a unique address that determines where it is. This address is normally assigned from a hierarchical system. For example, you can be "Fred Murphy" to people in your house, "Fred Murphy, 1 Main Street" to Dubliners, or "Fred Murphy, 1 Main Street, Dublin" to people in Ireland, or "Fred Murphy, 1 Main Street, Dublin, Ireland" to people anywhere in the world. On the Internet, addresses are known as Internet Protocol (IP) addresses.

    - Message forwarding

Since many networks are partitioned into subnetworks and connect to other networks for wide-area communications, networks use specialized hosts, called gateways or routers, to forward packets between networks. This is also of interest to mobile applications, where a user may move from one location to another, and it must be arranged that his messages follow him. Version 4 of the Internet Protocol (IPv4) was not designed with this feature in mind, although mobility extensions exist. IPv6 has a better designed solution.

Within the service layering semantics of the OSI network architecture, the network layer responds to service requests from the transport layer and issues service requests to the data link layer.[28]

Network Routing

Routing is the task of directing data flows from source nodes to destination nodes while maximizing network performance. This is particularly complicated in the network has dynamic nature, the topology can change constantly, and paths between sources and destinations that were initially efficient can quickly become inefficient or even infeasible. This means that routing information should be updated more regularly than in traditional wired telecommunication networks, and is a problem because the network has limited bandwidth and node resources, and their possibly unreliable communication channels.

Routing metrics

Routing algorithms need a metric to evaluate different routes and choose one (or several) among them. The most straightforward choices are: (a) the end-to-end delay and (b) the hop count.

End-to-End delay most times is unstable because the traffic of one neighbor can easily affect other neighbors and vice versa. Most times it happens because of the limited number of routes. When we have limited shortest paths limited wireless devices and small queues shared all connected nodes, AntHocNet prefer to pick those routes as a result of that causing congestion in the network.

Compared to delay, the hop count is a very simple and stable metric. Hop count is not necessarily a good metric [9]. This is because paths with a low number of hops usually consist of long hops, which can be of low quality and break easily as a consequence of node movement, and because short paths tend to go through the center of the AHWMN area, where congestion and wireless channel contention is higher.

A number of other metrics have been proposed to evaluate paths.

a) In Associativity-Based Routing (ABR), nodes periodically broadcast beacon messages, and they count how many of these messages they have received from each of their neighbors.

b) Links with a high number of associativity ticks are considered more stable, and are therefore preferred.

c) Power aware routing algorithms evaluate paths based on their power usage. Different power based metrics are possible. One can e.g. choose the path which uses minimal power, or the path going over nodes with most power left.

d) An interesting newly proposed metric is the Expected Transmission Count (ETX). This metric estimates the expected number of transmissions that will be needed to successfully conclude the transfer of a data packet over a link and it includes possible retransmissions of the data packet due to failures, and the transmission of the acknowledgment packet in backward direction.

Unicasting versus multicasting and broadcasting

In unicast routing protocols, a sender sends to one particular receiver. In multicast protocols, the group of receivers is a subset of the nodes of the network, where each member needs to be explicitly identified. Usually Multicasting tactic is used for route discovery as a flooding technical. Multicasting could be provided via the use of parallel unicast sessions between the source and each of the destinations. Such an approach would however be quite inefficient. A commonly used alternative is to built a routing structure between the members of the multicast receiver group, so that messages from the source can efficiently be forwarded between them. Finally, Broadcasting, has a group of receivers contains all the nodes in the network, also for periodical updates and routes maintaining process unicast and currently used nodes are communicating together.

Proactive versus reactive routing algorithms

Traditionally, AHWMN routing protocols are classified as proactive or reactive protocols. In reactive routing protocols, nodes only gather the routing information on-demand, which means only when they needed. This is the case when a new data session is started, or when a currently used path fails. Start of the reactive process occurs only when we need to send data packets, at the initiation of the session or when a route fails and we want new routing information for nodes . Reactive algorithm can greatly reduce the overhead they create, so that they are in general more efficient. The drawback of the reactive algorithms is that when routes are failing we do not have alternative routes. Here is occurring a delay while the protocol need to sent data packets to discover new routes to forward the data packets.

In case of proactive routing protocols all nodes of the network periodically are informed in case to maintain routing information about node and their current behavior. That means the nodes get periodically update which is effecting the throughput of the network and also causes end-to-end delay for the control packets. The advantage of the current algorithm usage is that there is no need of route discovery after a failure occurs cause nodes are maintaining routing information and can find really fast new routes. Also provides backup paths why most used and shortest paths fail or are heavily congested. On the downside, these algorithms can become quite inefficient or even break down completely when a lot of changes need to be tracked. This is the case when the topology is highly dynamic, or when the network is large[8].

Chapter 3

AntHocNet Algorithm

3.1 AntHocNet

AntHocNet is a hybrid, adaptive routing algorithm that combines both reactive and proactive routing. Specifically, it combines a reactive route setup process with a proactive route maintenance and improvement process. The way AntHocNet gathers, stores and uses routing information is inspired by the ACO approach to routing and from distance vector routing.

Next, we present an overview of the AntHocNet algorithm followed by a detailed examination of each of its component and also state diagrams for the algorithm.

3.2 General description of the algorithm and terminologies

AntHocNet contains both reactive and proactive elements. The algorithm is reactive in the way of sense that it only gathers routing information about destinations that are involved in communication sessions. It is proactive in the sense that it tries to maintain and improve information about existing paths while the communication session is going on. Routing information is stored in pheromone tables. Forwarding data packets is done in a stochastic way, using the information stored in the tables by an equation calculation. Two tables are used in this approach.

Pheromone tables:

Each node i maintains one pheromone table Ti, which is a two-dimensional matrix. An entry T_{ij}^{d} of this pheromone table contains information about the route from node i to destination d over neighbor j:

a) This information includes the pheromone value τ_{ij}^{d}, which is a value indicating the relative goodness of going over node j when traveling from node i to destination d.
b) Virtual Pheromone value

Neighbor table:

This table keeps track of the wireless nodes to which it has a wireless link and also information about last heard from this neighbor.

The algorithm is composed of two main parts the Reactive part and the Proactive part. Next is a description of the two parts of the algorithm. The algorithm is also depicted at all parts of Figure 3.

A) Reactive Route Setup

The reactive route setup process is triggered whenever a node receives a data packet that was logically generated for a destination for which no routing information is available. This lack of routing information can happen either because the data packet in question is the first of a new communication session, and no routing information for its destination is available from a different or previous session, or because the data packet belongs to an ongoing session for which all routes have become invalid. The reactive route setup process involves the sending of reactive forward ant from source to destination, and a reactive backward ant from destination to source. In this part there is a represents the reactive component of the algorithm. It starts at the beginning of a communication session.[12]

1) Source node of the data packet check neighbor table who can forward the data packet to the destination.
2) If no destination is available, it initiates a reactive route setup process, in which it sends an ant packet(control packets) out over the network to find a route to the destination. Such an ant packet is called a reactive forward ant. Basically it broadcast Ant Control Packets in all the network, a flooding tactic to explore the current network.
3) Each intermediate node that receives a copy of the reactive forward ant, forwards it and on receiving duplicate control packets are discarded immediately. This is done via broadcasting in case the node does not have routing information about the ant's destination in its pheromone table. If routing information is available the packet is unicast to the best neighbor(neighbor with highest pheromone).

Reactive forward ants:

1) As these ants travel through the network towards the sink (via broadcasting) they store the nodes that they have visited on their way in a list inside the control packet.
2) The first copy of the reactive forward ant to reach the destination is converted into a reactive backward ant, while subsequent copies are destroyed from the same source.

Reactive backward ant:

1) Retraces the exact path that was followed by the forward ant back to the source.
2) On its way, it collects quality information about each of the links of the path.
3) At each intermediate node and at the source, it updates the routing tables based on this quality information.

B) Proactive route maintenance process:

The proactive route maintenance process serves to update and extend available routing information. In particular, it allows to build a mesh of multiple routes around the initial route created during the reactive route setup process. The proactive route maintenance process consists of two sub processes : pheromone diffusion and proactive ant sampling. Pheromone diffusion is aimed at spreading available pheromone information over the network through the use of periodic update messages and information bootstrapping. Proactive ant sampling is aimed at controlling and updating pheromone information through path sampling using proactive forward ants. While proactive ant sampling is started by the source node of a communication session at the start of the session and continues for as long as the session is going on, pheromone diffusion is executed by all nodes throughout their whole lifetime, and is not particularly bound to the occurrence of single session. Below, there is step by step process of the algorithm.[30]

Pheromone diffusion:

1) It is the first sub-process of proactive route maintenance. It can be considered a cheap but unreliable way of spreading pheromone information.
2) Spreads out pheromone information that was placed by the ants.
3) Nodes periodically broadcast messages containing the best pheromone information they have available to create shortest paths in the network.
4) Using information bootstrapping, neighboring nodes can then derive new pheromone for themselves and further forward it in their own periodic broadcasts.

Proactive ant Sampling :

1) This sub-process turns the virtual pheromone into reliable regular pheromone.
2) All nodes that are the source of a communication session periodically send out proactive forward ants towards the destination of the session.
3) These ants construct a path in a stochastic way, choosing a new next hop probabilistically at each intermediate node.
4) Different from reactive forward ants, they are never broadcast.
5) When calculating the probability of taking a next hop, proactive forward ants consider both regular and virtual pheromone.
6) This way, they can leave the routes that were followed by previous ants, and follow the (potentially unreliable) routes that have emerged from pheromone diffusion.

7) Once a proactive forward ant reaches the destination, it is converted into a proactive backward ant that travels back to the source and leaves pheromone along the way (regular, not virtual pheromone), just like reactive backward ants.
8) Proactive ants can follow virtual pheromone and then, once they have experienced that it leads to the destination, convert it into regular pheromone.

Types of pheromones

In AntHocNet we have two different types of pheromones which are :

Virtual Pheromone

Virtual pheromone: is the pheromone that is obtained via pheromone diffusion and is kept separate from the normal pheromone placed by the ants because of its potential unreliability. It is obtained via diffusion and it is spread via proactive and sampling.

Regular Pheromone

Regular pheromone: is the reliable pheromone placed by the ants. Regular pheromone is acquired through the reactive backward ant when the reach destination and the current route is correct and reliable.

Data packet forwarding:

Data packets are forwarded from their source to their destination in hop-by-hop fashion , taking new routing decision at each intermediate node. Routing decisions for data packets are based only on regular pheromone and an mathematical equation. This means that they only follow reliable routes that are the results of ant based sampling, and leave the virtual pheromone information that is the result of information bootstrapping out of consideration. The combination of the reactive route setup process and the proactive route maintenance processes leads to the availability of a full mesh of such reliable routes between the source and destination of each session.

The gathered pheromone values are used for forwarding the data packet.

1) Routing decisions are taken hop-by-hop, based on the locally available pheromone.
2) Only regular pheromone is considered, as virtual pheromone is not considered reliable enough. If not regular pheromone is available virtual pheromone is selected.
3) Each forwarding decision is taken using a stochastic formula that gives preference to next hops that are associated with higher pheromone values.

Link Failure and Detection: can occur due to physical changes such as the movement or disappearance of a node, or due to changes that influence the connectivity of the wireless communication, such as an increase of radio interference or a decrease in the used transmission power. Link failure can be detected if protocol report the failure of the unicast transmission of a control or data packet, or if a node fails to receive periodic hello messages from its neighbors.

1) Each node sends to its neighbors at a fixed interval of tHello seconds.
2) When a node i receives msg from a new node j , it can assumes that j is its neighbor and create a Nij entity for its neighbor table.
3) After creating the neighbor entity it creates an entity Tij in its pheromone table indicating the next hop.
4) If i does not hear from j for a certain number of tHello seconds intervals, i assumes that the wireless connection to j has disappeared.
5) In AntHocNet tHello messages are not only used to detect link failures but also to deliver pheromone information for diffusion process.

Link Failure Notification: First reaction of the AntHocNet routing protocol on Link Failure Detection

1) First it removes the failure node from its neighbor table
2) Then, it updates its pheromone table Ti, building a link failure notification message in the process.
3) Then, it scans its pheromone table to control which destination d have a non-zero regular pheromone value Tdij.

4) Next, it checks whether the lost pheromone value Tdij was the best or the only regular pheromone value available for d.
5) If yes then it adds the address of the destination d to the link failure notification message, together with the new best regular pheromone it has available for d.
6) If Tdij was the only non-zero regular pheromone entry for d, this is also indicated in the link failure notification message.
7) Once the link notification message is fully constructed, it is broadcast.
8) All nodes receiving the notification message updates their neighbor tables.

Local Route Repair Process: in this procedure AntHocNet Routing protocol tries to repair broken routes via repair forward ant. This process is more likely the ant reactive route setup process with a limited maximum broadcasting per node.

1) When Link Failure notification message is received from the nodes the node which detected the link failure start ant repair forward process, where it broadcasts control packet to search for new routes to reach the sink.
2) Upon arrival at the destination d, repair forward ant is converted into a repair backward ant, which travels back to the node i that launched the local route repair process.
3) Traveling back to the source , updating regular pheromone entries on the way.
4) Once the repair backward ant is back at the original node i, node i send its stored pending data packets to destination d.
5) Node i which has broadcast the link failure notification message has a timer which it starts after the repair forward ant process initiated, if no repair backward ant received before the timer runs out stored data packets discarded and broadcast link failure notification about destination d.

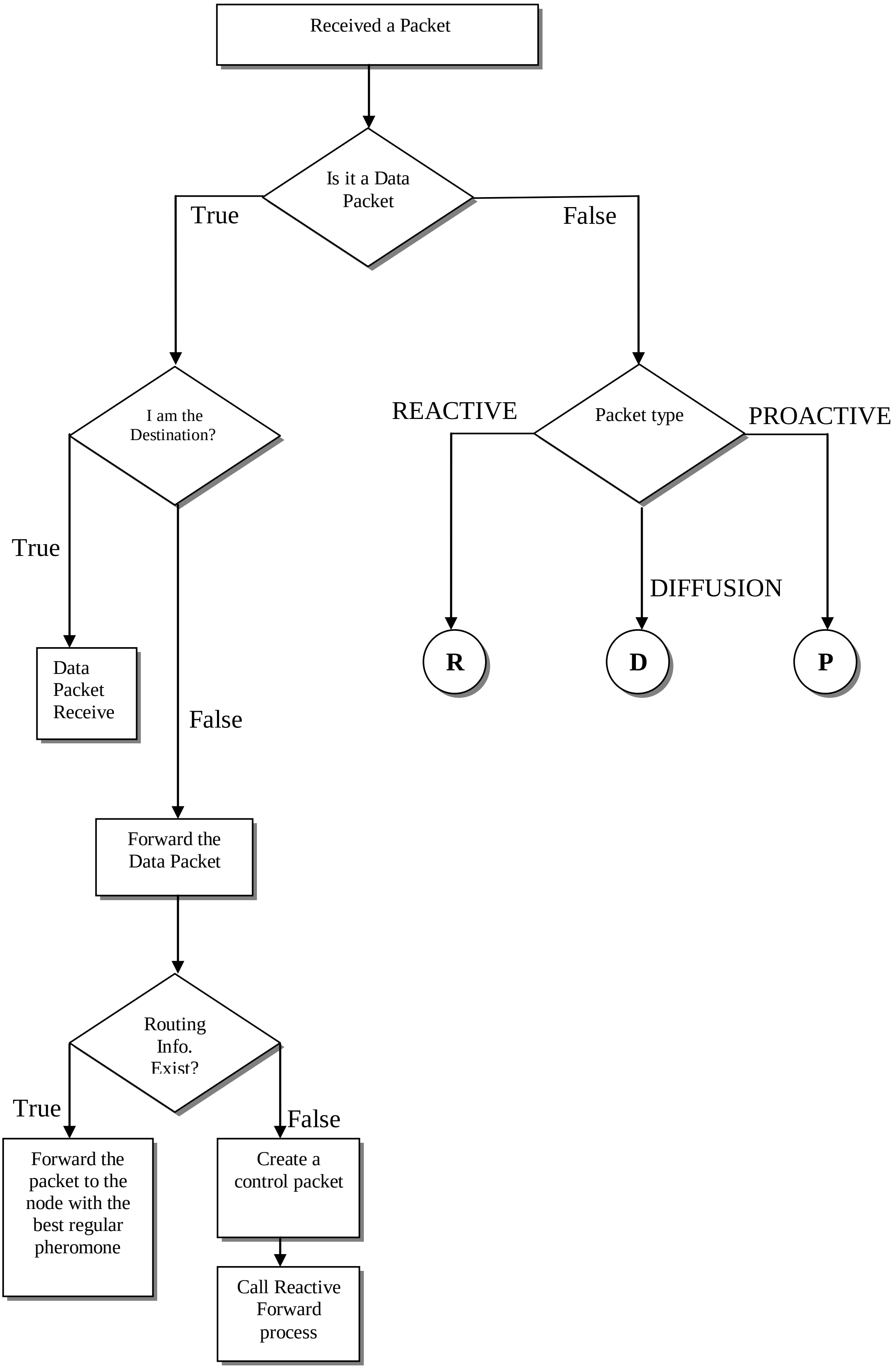


Figure 3.1: The algorithm of the process when a packet is received

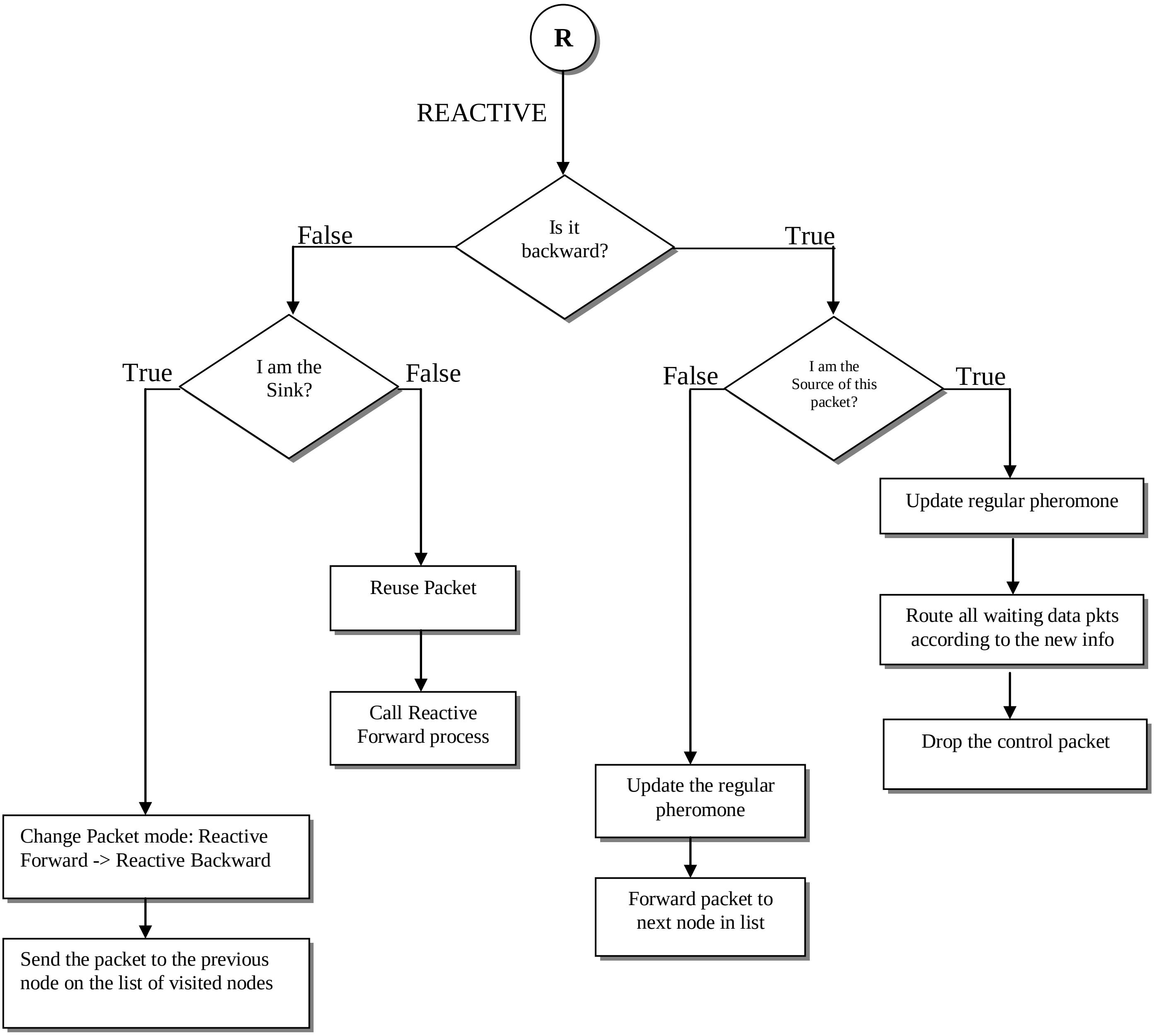

R
REACTIVE
Is it backward?
False
True
I am the Sink?
True
False
I am the Source of this packet?
False
True
Update regular pheromone
Reuse Packet
Route all waiting data pkts according to the new info
Call Reactive Forward process
Drop the control packet
Update the regular pheromone
Change Packet mode: Reactive Forward -> Reactive Backward
Forward packet to next node in list
Send the packet to the previous node on the list of visited nodes

Figure 3.2: The algorithm of the process when the packet received is reactive ant packet.

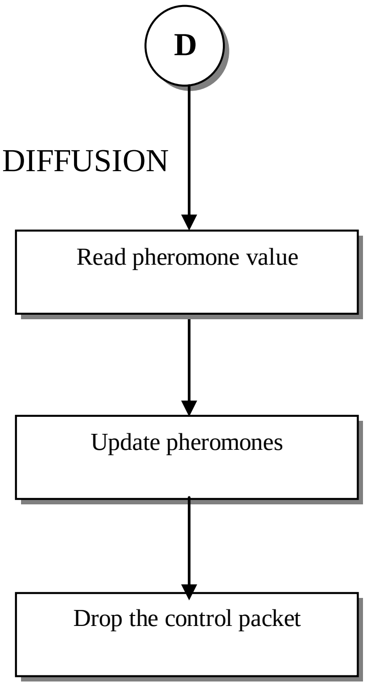


Figure 3.3: The algorithm of the process when the packet received is diffusion ant packet.

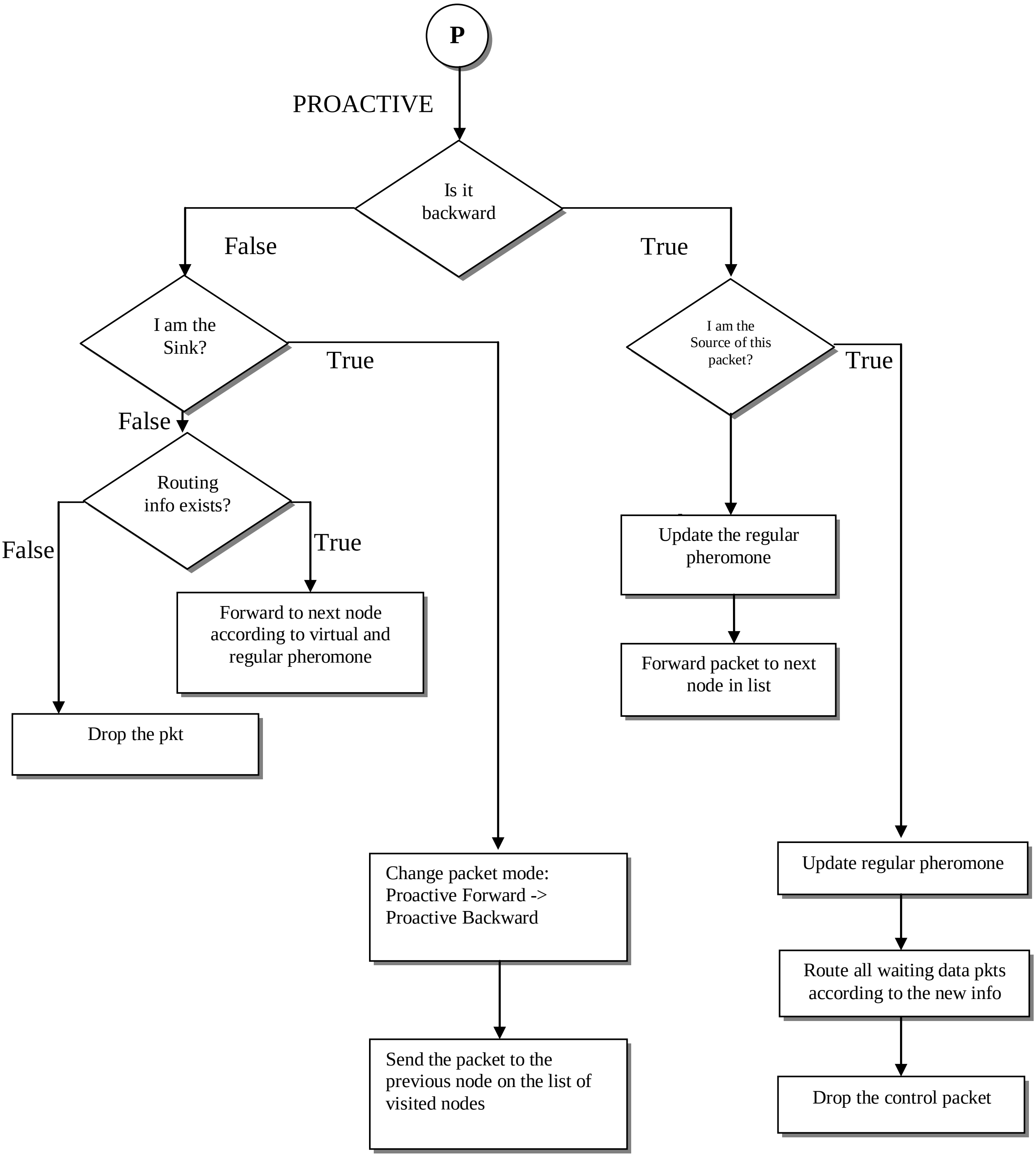

P
PROACTIVE
Is it backward
False
True
I am the Sink?
True
False
Routing info exists?
False
True
Forward to next node according to virtual and regular pheromone
Drop the pkt
Change packet mode: Proactive Forward -> Proactive Backward
Send the packet to the previous node on the list of visited nodes
I am the Source of this packet?
True
Update the regular pheromone
Forward packet to next node in list
Update regular pheromone
Route all waiting data pkts according to the new info
Drop the control packet

Figure 3.4: The algorithm of the process when the packet received is proactive ant packet.

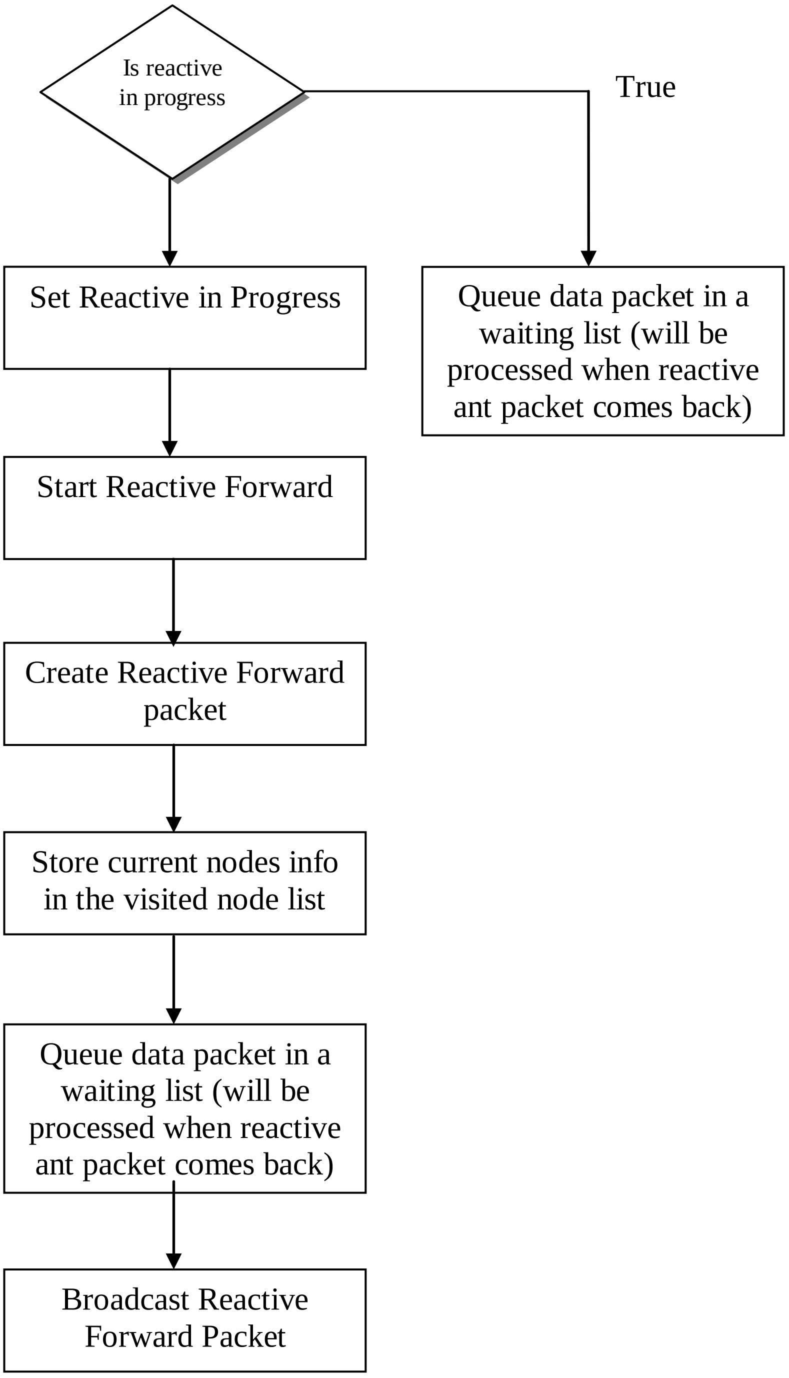

Is reactive in progress
True
Set Reactive in Progress
Queue data packet in a waiting list (will be processed when reactive ant packet comes back)
Start Reactive Forward
Create Reactive Forward packet
Store current nodes info in the visited node list
Queue data packet in a waiting list (will be processed when reactive ant packet comes back)
Broadcast Reactive Forward Packet

Figure 3.5: The algorithm of the Reactive process

Link Failure Detection and Repairing

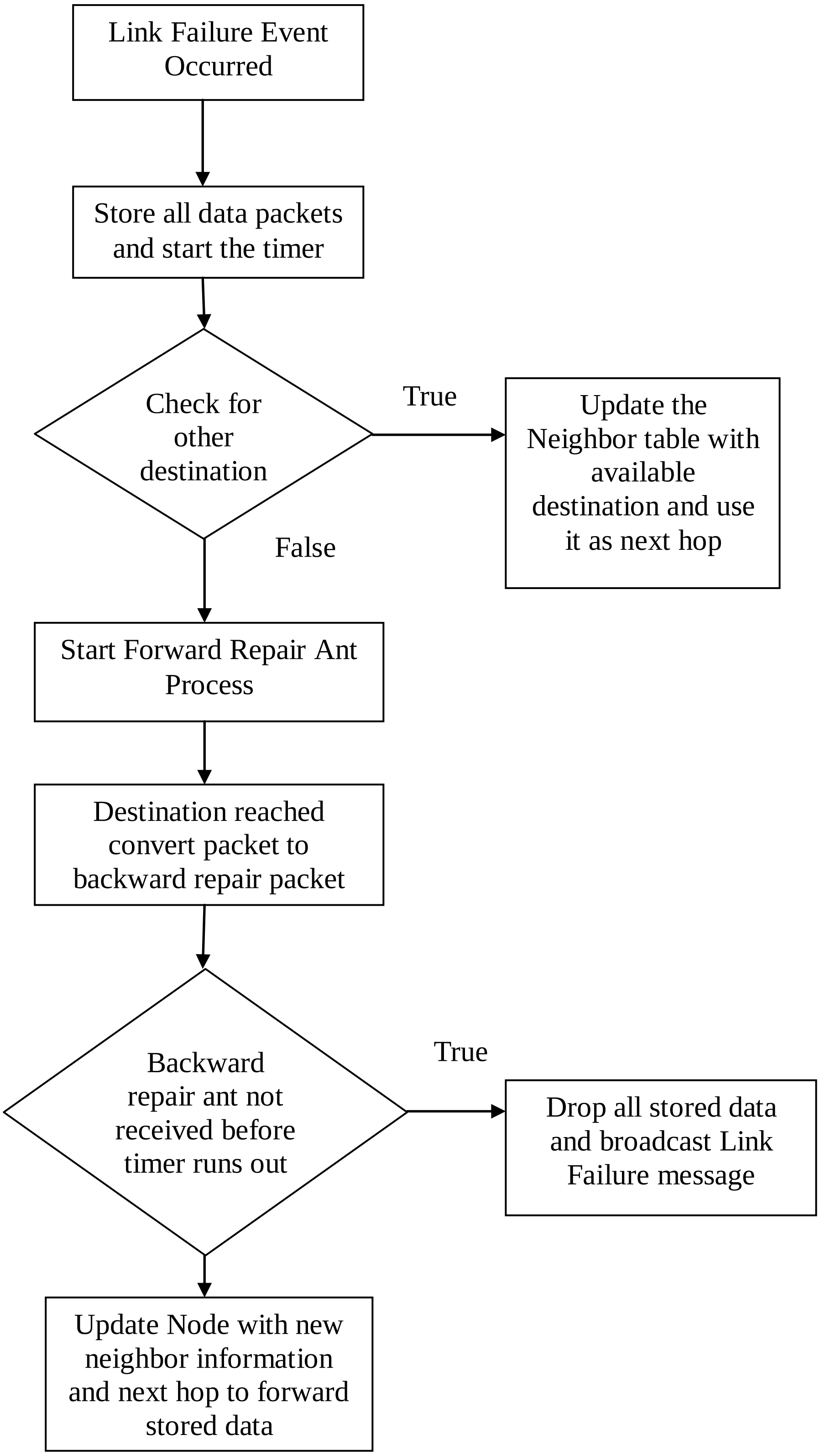

Link Failure Event Occurred
Store all data packets and start the timer
Check for other destination
True
Update the Neighbor table with available destination and use it as next hop
False
Start Forward Repair Ant Process
Destination reached convert packet to backward repair packet
Backward repair ant not received before timer runs out
True
Drop all stored data and broadcast Link Failure message
Update Node with new neighbor information and next hop to forward stored data

Figure 3.6: The algorithm Link Failure Route Repair

3.3 Detailed description of the algorithm

In this section I will provide farther information about the algorithm and more specifically about the data structure of the algorithm based on the one presented in [8] and [12].

Data structures in AntHocNet

General parameter description : i descript as the current node
d descript as the sink node
j descript as the current node next best hop

Pheromone table:

Each node i maintains a pheromone table Ti, which is a two dimensional matrix. 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) a regular pheromone value τ_{ij}^{d},
b) virtual pheromone value ω_{ij}^{d},
c) an average number of hops h_{ij}^{d}.

Regular pheromone value τ_{ij}^{d} is an estimate of the goodness of the route from i(current node) to d (sink node) over j(best next neighbor). Goodness is expressed as the inverse of a cost by an equation descript in farther topics. Regular pheromone is updated by backward ants while returning to the source nodes. These can be reactive, proactive or repair backward ants for route verification.

Virtual pheromone value ω_{ij}^{d} forms an alternative estimate of the goodness of the route from i to d over j. Virtual pheromone is obtained through information bootstrapping using goodness values reported by neighbor nodes during the proactive route maintenance process.

Average number of hops h_{ij}^{d} is like the regular pheromone, updated by backward ants. h_{ij}^{d} is used when deciding how long to wait for repair backward ants.[24]

Neighbor table:

Each node i maintains a neighbor table N_i kept by node i as a one-dimensional vector with one entry for each of i' s neighbors. The entry N_{ij} corresponding to i' s neighbor j contains a time value th_{ij} that indicates when i last heard from j. Node i uses this time value to derive whether there is a wireless link with node j, and to detect link failures. If node i haven't heard thello message from neighbor j after an interval of time node j is considered as an dead node and starts ant repair forward process.

Important Notes:

Nodes do not necessarily always have values available for τ_{ij}^{d}, ω_{ij}^{d}, h_{ij}^{d} because nodes do not maintain routing information about all possible destinations in the network and because for a specific destination, nodes do not necessarily have a route available over each of their possible neighbors. Since τ_{ij}^{d} and h_{ij}^{d} are both obtained from backward ants, nodes that have a value for one of the two will also have a value for the other. Since regular and virtual pheromones are obtained through different processes, it is possible that a node has a value for τ_{ij}^{d} but not for ω_{ij}^{d} or vice versa.[30]

The approach of keeping virtual and regular pheromone separate means that bootstrapped pheromone is not used directly for the forwarding of data packets, since data packets only consider regular pheromone when choosing a next hop. Virtual pheromone is used when forwarding proactive forward ants towards their destination.[24]

Reactive and proactive ants follow similar procedures with small differences. Unlike reactive forward ants, proactive forward ants rely both on regular and virtual pheromone for their routing decisions: they use the maximum between regular and virtual pheromone to calculate the probability of each next hop. Also different from reactive forward ants is that proactive forward ants are never broadcast: when they arrive at a node that does not have any routing information for their destination, they are discarded.[24]

Reactive route setup:

In case there is no available information while source nodes are trying to send data packets the AntHocNet immediately triggers the reactive route setup process in case to find new routes. This procedure is the route discovery process of AntHocNet. While no information is available which means the network just now initiated the data packet forwarding or a route failure may occurred. The main function of Reactive route setup process is to make a flooding in the network from ant packets which in our case are considered the control packets. The creation of each control packet is done at the source nodes and with destination the Sink and source node the current node who created the packet. After the creation the broadcast process takes place. Every node who already received the control packet before discarded it. Reactive control ants are searching their way to the sink through the whole network.

After the forward reactive ant packets reaches the sink duplicates are drop and the first packets received are converted to reactive backward ants. Reactive backward ants following the path list of the packet travels back to the source nodes in case to update the nodes pheromones. After reaching the source reactive backward ants are dropped and pending data packets are forwarded to the sink node and the data packet process is continue normally.

<u>Reactive forward ants: (This part is form reference [24])</u>

1) At the start of the reactive route setup process, the source node s creates a reactive forward ant. This is a control packet that has as a goal to find a path from s to an assigned destination d, but to flood in the network. During this process all data packets create by the traffic agent will be stored.

2) At the start, the ant contains just the addresses of s and d. Later, as it proceeds through the network, it collects a list P = [1, 2, …, d - 1] of all the nodes that it has visited on its way from s to d.

3) After its creation at s, the reactive forward ant is broadcast, so that all of s's neighbors receive a copy of it. At each subsequent node, the ant is either unicast or broadcast, depending on whether the current node has routing information for d.

4) If routing information is available, the node chooses a next hop for the ant probabilistically, based on the different pheromone values associated with next hops for d.

$$P_{in}^{d} = \frac{(\tau_{in}^{d})^{\beta 1}}{\sum j \in N_i^d \, (\tau_{ij}^{d})^{\beta 1}}$$

$\beta 1 >= 1$

- node i
- chooses node n as next hop
- probability a node chose the next hop for the ant P_{in}^{d}
- N_i^d set of neighbors of i over which a path to d is known
- $\beta 1$ is a parameter value which can control the exploratory behavior of the ants (on 20, relatively high because we want to obtain the initial route as fast as possible, and limit the time we spend on exploration at this stage)

5) In case the intermediate node i does not have routing information for d, it broadcasts the reactive forward ant. Due to this broadcasting, a reactive forward ant can build up quickly over the network, with different copies of the ant following different paths to the destination.

6) In order to limit this overhead, nodes only forward the first copy of the ant that they receive. Subsequent copies are simply discarded.

7) At the destination, the reactive forward ant is converted into a reactive backward ant, which follows the list of nodes P visited by the forward ant back to s.

8) If more than one copy of the forward ant is received, only the first is accepted and converted into a backward ant, while subsequent copies are discarded. This way, only one route is set up during the reactive route setup process.

Reactive backward ants(This part is form reference [24])

1) The reactive backward ant created by the destination node in response to a reactive forward ant contains the addresses of the forward ant's source node s and destination node d, as well as the full list of nodes P that the forward ant has visited.

2) The reactive backward ant is unicast from d and among the nodes of P back to s.

3) The aim of the reactive backward ant is to update routing information in each of the nodes of P and in s. At each node i that it visits, it updates the number of hops and the regular pheromone value in the pheromone table entry $\mathbf{T_{ij}^{d}}$, where n is the node that it visited before i on its way from d.

4) The updating of number of hops is done using a moving average.

$$\mathrm{h_{in}^{d}} \leftarrow \mathrm{ah_{in}^{d}} + \ (1 - \mathrm{a})\mathrm{h}$$

$$a \in [0,1]$$

- h is the number of hops that the backward ant has traveled between d and i

- α is a parameter regulating how quickly the formula adapts to new information (it is kept on 0.7)

5) Updating of the regular pheromone is done based on the cost of the route from i to d. This cost can be calculated using different metrics, such as the number of hops, the end-to-end delay, etc.. Here, we talk in terms of a generic cost $\boldsymbol{c_i^j}$, where is the cost of the link from i to its neighbor j (it is the cost count from the number of hop). Under AntHocNet, each node maintains a local estimate of the cost $c_i^j$to go to each of its neighbors j.

6) The reactive backward ant reads at each node i the local estimate c_i^n of the cost to go from i to the next hop n that the ant is coming from.

7) The reactive backward ant adds this cost to the total cost c of the route from n to d (which it has been calculating on its way back from d), which is stored inside the ant.

8) The new cost c_i^d is used to update the pheromone value in node i, using the moving average formula of equation

$$\tau_{ij}^d \leftarrow \gamma\tau_{ij}^d + \ (1-\gamma)(c_i^d)^{-1}$$

$$\gamma \in [0,1]$$

- γ is a parameter regulating the speed of adaptation of the pheromone to new cost values. (γ was kept on 0.7)
- The cost value c_i^d is inverted to calculate the pheromone value τ_{ij}^d, as pheromone indicates the goodness of a route, rather than its cost.

Proactive route maintenance

The proactive route maintenance process serves to update and extend available routing information. In particular, it builds a mesh of multiple routes around the initial route. The proactive route maintenance process consists of: pheromone diffusion and proactive ant sampling. Pheromone diffusion is aimed to spread all this pheromone information over the network through the use of periodic update messages and information bootstrapping, so that a field of pheromone pointing towards the destination becomes available in the network. This field of pheromone is indicated in the virtual pheromone values in the pheromone tables of the nodes. The fact that pheromone is spread out is similar to the normal diffusion of real pheromone in nature, which allows ants further away to sense it. [30]

The pheromone diffusion is executed by all nodes throughout their whole lifetime, and is not particularly bound to the occurrence of a single session. Node with no pheromone information will not take part in the process in case to reduce networks throughput and congestion.[30]

Proactive ant sampling is aimed at controlling and updating pheromone information through path sampling using proactive forward ants, in case of finding new routes shortest for the data packets. [30]

The proactive ant sampling is started by the source node of a communication session at the start of the session and continues for as long as the session is going on. Source nodes are creating proactive forward ants and forward then in the network. This process is similar to the Reactive route setup process.[30]

Pheromone diffusion(from reference [30])

1) Essential and crucial role in the pheromone diffusion process is played by hello messages. These are short and small load messages broadcast every t_{hello} seconds asynchronously by all the nodes of the network which are maintain routes. In AntHocNet, $\boldsymbol{t_{hello}}$ is set to 1 second. t_{hello} messages allow nodes to find which are their immediate neighbors so they can detect link failures and convey routing information inside these hello ant messages. While also in AntHocNet hello messages serve as the periodic update messages that are needed in the information bootstrapping process of pheromone diffusion. Also this messages are important because are used for link failures detection.

2) Nodes include in each hello message that they send out routing information they have available. In particular, a node i constructing a hello message consults its pheromone table, and picks a maximum number k of destinations it has routing information for.

3) For each one of these destinations d, the hello message contains the address of d, the best pheromone value that i has available for d, v_i^d and a bit flag.

4) This best pheromone value v_i^d is taken over all possible values for regular pheromone and virtual pheromone associated with d in i's pheromone table Ti.

5) The bit flag is used to indicate whether the reported value was originally regular or virtual pheromone.

6) A neighboring node j receiving the hello message from i goes through the list of reported destinations. For each listed destination d, it derives from the hello message an estimate of the goodness of going from j to d over i, by applying information bootstrapping: it combines the reported pheromone value , which indicates the goodness of the best route from i to d, with the locally maintained estimate of the cost c of hopping from j to i.

$$\kappa_{ji}^{d} = ((v_i^d) + c_j^i)^{-1}$$

- v_i^d goodness value
- c_j^i cost value
- $\kappa_{ji}^{d}$the result of the calculation is what we call the bootstrapped pheromone value

7) With κ_{ji}^{d}, node j has obtained a new estimate for the goodness of the path to d over i in a relatively cheap way. Thanks to the use of information bootstrapping, all that was needed in terms of communication overhead was the sending of the value v_i^d from i to j.

8) Node j maintains in its pheromone table entry two distinct pheromone values for the route over its neighbor i to destination d: the regular pheromone value and the virtual pheromone. Of these, only the virtual pheromone value is normally updated with the new bootstrapped pheromone value κ. This way, the pheromone obtained via the pheromone diffusion process is kept separate from the regular pheromone, which is the product of ant based route sampling and is therefore considered more reliable. The updating is done by replacing ω by κ.

9) If the regular pheromone of a node is empty and it has no information, the regular pheromone is obtained via the pheromone diffusion process. This updating replacement improves our algorithm because instead of no information it has a less reliable information stored. Furthermore we keep a flag so with the first chance to update the regular pheromone with reliable information.

10) When reaching the destination, proactive forward ants are converted into proactive backward ants, which do deposit regular pheromone, which in turn is used for routing data packets. So, in this way, bootstrapped pheromone influences data forwarding indirectly. One could say that the potentially

unreliable bootstrapped pheromone provides hints about possible routes, which are then explored and verified by the proactive forward ants.

11) There is one situation that forms an exception to this normal mode of operation, in which we do allow the bootstrapped pheromone value κ to be used for updating the regular pheromone value τ and influencing data forwarding directly. This is the case when the following two conditions are fulfilled:
   a) j already has a non-zero value for the regular pheromone τ
   b) the bootstrapped pheromone κ was derived from a reported pheromone value v that was based on regular pheromone in i, rather than virtual pheromone .

12) When a node receives a diffusion, hello message packet it update its tables with the new information and it deletes the ant packet.

Proactive ant sampling(from reference [30])

1) The proactive ant sampling process is started by the source node of a session at the moment the first data packet of a new session is received, and continues for as long as the session is going on. The aim of the process is to use ant based sampling to gather routing information for ongoing sessions.

2) To this end, proactive forward ants are generated. These ants can follow regular pheromone, which is routing information placed by previous ants, or virtual pheromone, which is the result of the pheromone diffusion process described above.

3) While the former leads the ants to update goodness estimates of existing routes, the latter allows them to find new routes based on the hints provided by the pheromone diffusion process. This way, the single route that was initially constructed in the reactive route setup process is extended to a full mesh of multiple paths.

4) Each node which is the source of a communication session periodically (normally, we use a period of t_{hello} seconds) schedules the transmission of a proactive forward ant towards the session's destination.

5) In order to improve efficiency, the actual sending of a proactive forward ant is conditional to the availability of good new virtual pheromone: only if the best virtual pheromone is significantly better than the best regular pheromone, a proactive forward ant is sent out. In our experiments: we have found that the best threshold to use was: at least 20% better. This avoids congestion due to overloading the network with proactive forward packets, which occurred at smaller threshold values like 5% or 10%.

6) The aim of the proactive forward ant is to find a route towards the destination, and to store the list of nodes P that it visits on the way. The proactive forward ant takes a new routing decision at each intermediate node i, using the formula of equation to calculate the probability of choosing each possible next hop n.

$$P_{in}^{d} = \frac{[\max(\tau_{in}^{d}, \omega_{in}^{d})]^{\beta 2}}{\sum_{j \in N_i^d}[\max(\tau_{ij}^{d}, \omega_{ij}^{d})]^{\beta 2}}$$

- β2>=1
- the function max(a,b) takes the maximum of the two values a and b
- β2 is a parameter that deffnes the exploratory character of the ants (is normally kept 20)
- list of nodes P that it visits on the way
- node i
- next hop n

7) When a proactive forward ant arrives at its destination, it is converted into a proactive backward ant, which is sent back to the source. Proactive backward ants have the same behavior as reactive backward ants:
   - follow the exact path P recorded by their corresponding forward ant back to the source
   - update regular pheromone entries at intermediate nodes and at the source

8) An important aspect to note here is that while the proactive forward ants can follow both regular and virtual pheromone, proactive backward ants always deposit regular pheromone. This way, the proactive ant sampling process can investigate promising virtual pheromone, and if the investigation is successful turn it into a regular route that can be used for data.

Data packet forwarding(from reference [30])

1) Data packets are forwarded from their source to their destination in hop-by-hop fashion, taking a new routing decision at each intermediate node.

2) Routing decisions for data packets are based only on regular pheromone. This means that they only follow the reliable routes that are the result of ant based sampling, and leave the virtual pheromone information that is the result of information bootstrapping out of consideration.

3) The combination of the reactive route setup and the proactive route maintenance processes leads to the availability of a full mesh of such reliable routes between the source and destination of each session. Nodes in AntHocNet forward data packets stochastically, based on the relative values of the different regular pheromone entries they have available for the packet's destination.

$$P_{dn} = \frac{(\tau_{in}^{d})}{\sum_{j \in N_{i}^{d}} (\tau_{ij}^{d})^{\beta 3}}^{\beta 3}$$

- β3>=1
- β3 for the power function of the pheromone values (is normally set 20)
- probability P for a node i to pick next hop n when forwarding a packet with destination d
- node i
- next hop n
- destination d

4) By adapting the β3 parameter, one can make data forwarding less or more greedy with respect to the best available routes.

   By setting β3 low, data is spread over multiple routes, considering also low quality ones. Using multiple routes for data forwarding can improve throughput, as the data load is spread more evenly over the available network resources.

   By setting β3 high, on the other hand, data is concentrated on the best routes. This can also be a good choice, since the routes that according to the ant sampling give the best performance, are exploited as much as possible.

Link Failure Notification and Local Route Repair

1) When Link Failure notification message is received from the nodes the node which detected the link failure start ant repair forward process, where it broadcasts control packet to search for new routes to reach the sink.

2) Upon arrival at the destination d, repair forward ant is converted into a repair backward ant, which travels back to the node i that launched the local route repair process.

3) Traveling back to the source , updating regular pheromone entries on the way.

4) Once the repair backward ant is back at the original node i, node i send its stored pending data packets to destination d.

5) Node i which has broadcast the link failure notification message has a timer which it starts after the repair forward ant process initiated, if no repair backward ant received before the timer runs out stored data packets discarded and broadcast link failure notification about destination d.

$$\text{timer} = 2 * \text{thop} * \text{h}h_{i}^{d}j$$

3.4 AntHocNet Parameters[24]

In this section we present a brief recap of the parameters controlling the inner work of the AntHocNet algorithm. This equations are the same in paper [30].

β1 is a parameter value which can control the exploratory behavior of the ants (on 20, relatively high because we want to obtain the initial route as fast as possible, and limit the time we spend on exploration at this stage)

$$\mathrm{P}_{\mathrm{in}}^{\mathrm{d}} = \frac{(\tau_{\mathrm{in}}^{\mathrm{d}})^{\beta 1}}{\sum \mathrm{j} \in \mathrm{N}_{\mathrm{i}}^{\mathrm{d}} (\tau_{\mathrm{ij}}^{\mathrm{d}})^{\beta 1}} \qquad \beta 1 >= 1$$

β2 is a parameter that defines the exploratory character of the ants (is normally kept 20)

$$\mathrm{P}_{\mathrm{in}}^{\mathrm{d}} = \frac{[\max (\tau_{\mathrm{in}}^{\mathrm{d}}, \omega_{\mathrm{in}}^{\mathrm{d}})]}{\sum_{\mathrm{j} \in \mathrm{N}_{\mathrm{i}}^{\mathrm{d}}} [\max (\tau_{\mathrm{ij}}^{\mathrm{d}}, \omega_{\mathrm{ij}}^{\mathrm{d}})]^{\beta 2}}^{\beta 2} \qquad \beta 2 >= 1$$

β3 for the power function of the pheromone values (is normally set 20)

$$\mathrm{P}_{\mathrm{dn}} = \frac{(\tau_{\mathrm{in}}^{\mathrm{d}})}{\sum_{\mathrm{j} \in \mathrm{N}_{\mathrm{i}}^{\mathrm{d}}} (\tau_{\mathrm{ij}}^{\mathrm{d}})^{\beta 3}}^{\beta 3} \qquad \beta 3 >= 1$$

α is a parameter regulating how quickly the formula adapts to new information (it is kept on 0.7)

$$\mathrm{h}_{\mathrm{in}}^{\mathrm{d}} \leftarrow \mathrm{a}\mathrm{h}_{\mathrm{in}}^{\mathrm{d}} + (1 - \mathrm{a})\mathrm{h} \qquad \mathrm{a} \in [0,1]$$

γ is a parameter regulating the speed of adaptation of the pheromone to new cost values. (γ was kept on 0.7)

$$\tau_{ij}^{d} \leftarrow \gamma\tau_{ij}^{d} + (1-\gamma)(c_{i}^{d})^{-1} \qquad \gamma \in [0,1]$$

Chapter 4

Evaluation

4.1 Evaluation

In this chapter there is a description about the evaluation of the implemented AntHocNet protocol, as a number of tools that I've used to take results and the whole procedure including the scenarios and some of the parameters I have used. In this section there are results of a large set of simulations, in which we introduce variations to the parameters controlling the inner behavior of the protocol under a number of different scenarios. Moreover, an analyze the effect of the different components of the AntHocNet and finally we compare the performance of the AntHocNet protocol with the FlockCC protocol under three different scenarios and data rates. For this comparison I picked the best parameters for AntHocNet and from FlockCC for each scenarios.

I have used the ns-2 network simulator which facilitated a fair comparison with existing algorithms (in particular the FlockCC).

4.2 Setup

As it is both expensive and complex to build a real AHWMN testbed for research, it is more convenient to use a simulator for the evaluation of research in this area. Moreover, a simulation makes it much easier to control all of the parameters involved (which might not be always possible in a real environment) and perform a large number of repeatable tests and low cost rather than using real sensors. Because ns-2.31 is an old version of network simulator I setup Fedore Core 2 using virtual machine VMWare Workstation 8 and through Fedora I was testing the protocol while transferring the results to Eclipse to visualize them.(visualization program created from Dr.Pavlos Antoniou).

4.3 The NS-2 Simulator

A number of different network simulator software packages are available to the research community. These include ns-2 [19]. To achieve the conclusion of thesis thesis project my implementation is created through ns-2.31 simulation as FlockCC routing protocol. Ns simulator usually is used for ad hoc net, and includes numbers suite of protocol and evaluations. This encouraged our use of the simulator in addition to the fact that it is free and open source. Moreover, the implementation and evaluation of the protocol we are comparing against (FlockCC) have been performed in this simulator, which makes the comparison between the two protocols more fair.

4.4 The Simulation Scenarios

NS-2 utilizes a special TCL script is used to instantiate the main simulator object and create the elements of the simulation scenario. The term scenario refers to a specific definition of the set of the variables affecting all aspects of the simulation. The main elements of the scenario include: the properties of the nodes and the events that will take part during the simulation.

The main properties of the nodes include: the number of the nodes (the size of the network), the displacement and mobility of the nodes (stationary or on the move), wireless range, energy model, queue type, the rate of data traffic, the routing protocol supported by the nodes and the internal parameters of the selected routing protocol time of the simulation nodes data initiation time and etc.

The events of the simulation scenario are actions that occur at specific moments in during the simulation time and mainly include: the registration/connection of the node in the network (starting of the node), the sending of data, failing of the nodes (injected failure of the nodes), disconnection of the nodes (stopping of the node). Numerous scenarios where created on this parameters to test and compare different algorithms on many occurring events.

All scenarios are using the same topology of nodes and positioning. Number of nodes are 300 displaced in a lattice of 20 rows and 15 columns.The distance between the node was set to 50 meters. For all the scenarios one node is designated as the destination of all the data packets (we refer to this node as the sink) and a number of nodes are chosen to be the data sending nodes (each call a source). Each such node starts a data session and sends a number of packets every second. The rate the data packets are sent is called the Constant Bit Rate (CBR). Each of the nodes implements the AntHocNet routing protocol and is capable of point-to-point or broadcasting of data or control packets. Usually data packets are unicasted and control packets unicasted and broadcasted(depends on the procedure we are using). For a specific scenario all the nodes have the same properties(queues, energy reserves, transmitting range etc). For the simulation of radio propagation, we used the two-ray signal propagation model (commonly used for modeling the propagation of wireless signals in open space [18]). The two-ray model assumes receiver gets the signal over two paths: a direct one and one reflected over the ground. At the physical layer, we use the IEEE 802.11 protocol. At the MAC layer, we use the CSMA IEEE 802.11 protocol with 2 Mbps/s transmission rate. We assume a radio range of 50m. For the transport layer, we use the UDP protocol.

Scenario Description(i used the same scenarios as Sandy in her thesis)

We have used three different scenarios for our evaluations. The first scenario (scenario 1) has 10 source nodes at the back of the network sending packet for the sink. All the source nodes start sending the data around the 10th second of the simulation. This resembles a quite common case where an external stimuli causes the triggering of nearby nodes at the same time. The second scenario (scenario 2) has 10 source nodes, with 5 of them placed at the back of the network and 5 at the front. The first 5 nodes start sending at the 10th second and the second 5 nodes start sending at the 50th second. At the 70th second the first 5 nodes stop sending data packets. The third scenario (scenario 3) is similar to scenario 1, however at the 40th second of the simulation a large number of nodes (between the source nodes and the sink) fail. All the scenarios ran for 100 seconds. For each of the scenarios we use thee different (rather high) CBR values in our simulations: 25 packet/second, 35 packet/second and 45 packet/second. These CBR rates represent slightly congested, congested and heavily congested cases of network traffic. The size of the network, duration of simulation, and the CBR value that we select for the simulation represent highly demanding scenarios of usage for the network and thus can be considered a real indicator of the performance of the protocols simulated at hand. Figure 7 illustrates the 2D spatial displacement of the nodes in each scenario.[24]

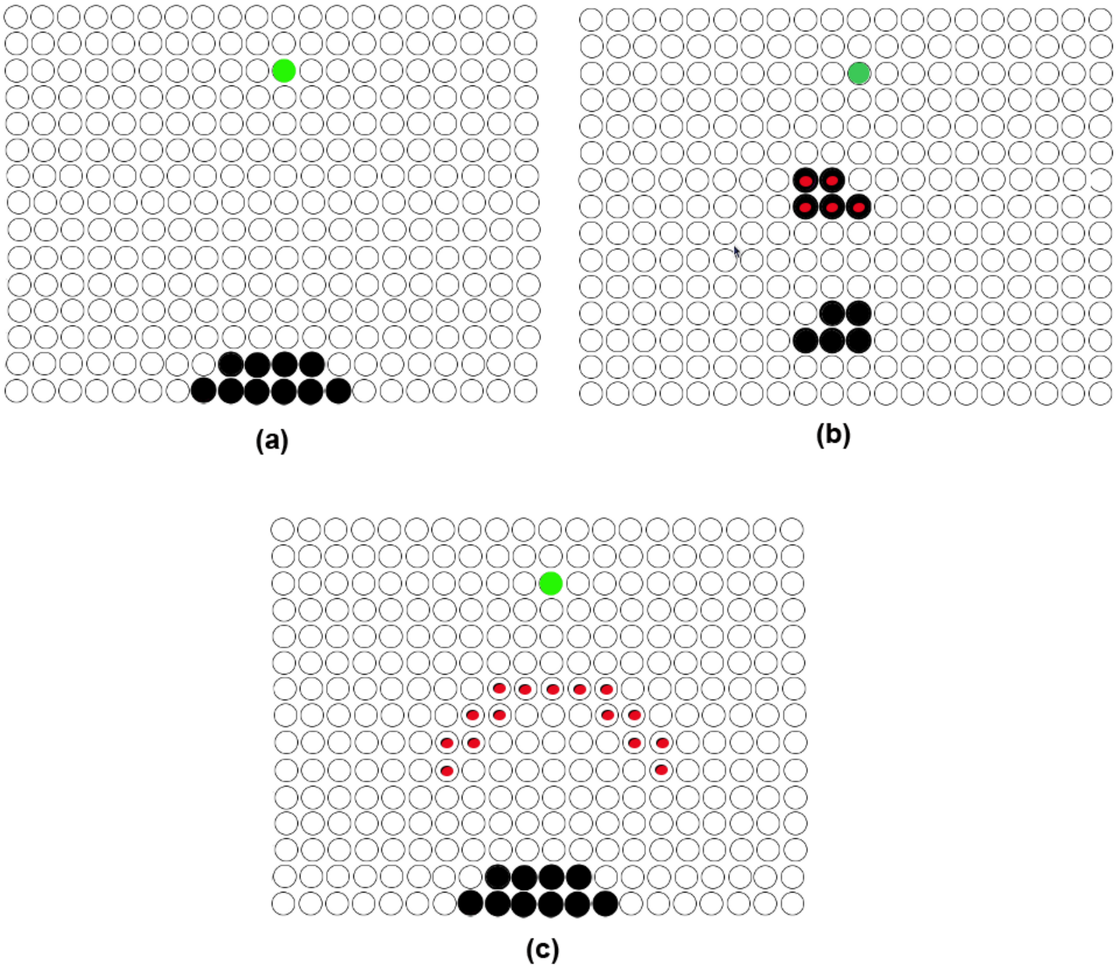


Figure 7: 2D spatial placement of the nodes in (a) scenario 1 (b) scenario 2 (c) scenario 3. Sender nodes are shown in black. Sink nodes are shown in green. Failing or deactivated nodes are shown with a red center.

4.5 Evaluation metrics

In this section, we describe the metrics we use to evaluate the performance of the routing algorithms. The first metric is the data delivery ratio. This is the ratio of correctly delivered data packets out of the total sent packets. Remember that in AHWMNs, due to the dropping of part of the packets (due to congestion for example) it might not be possible to deliver all the data packets. The second metric is the end-to-end packet delay. This is the accumulation of the delays experienced by packets traveling between their source and destination. The third metric is the Mean Energy which measures the amount of energy (in Joul) dissipated by all the nodes during the simulation.[24]

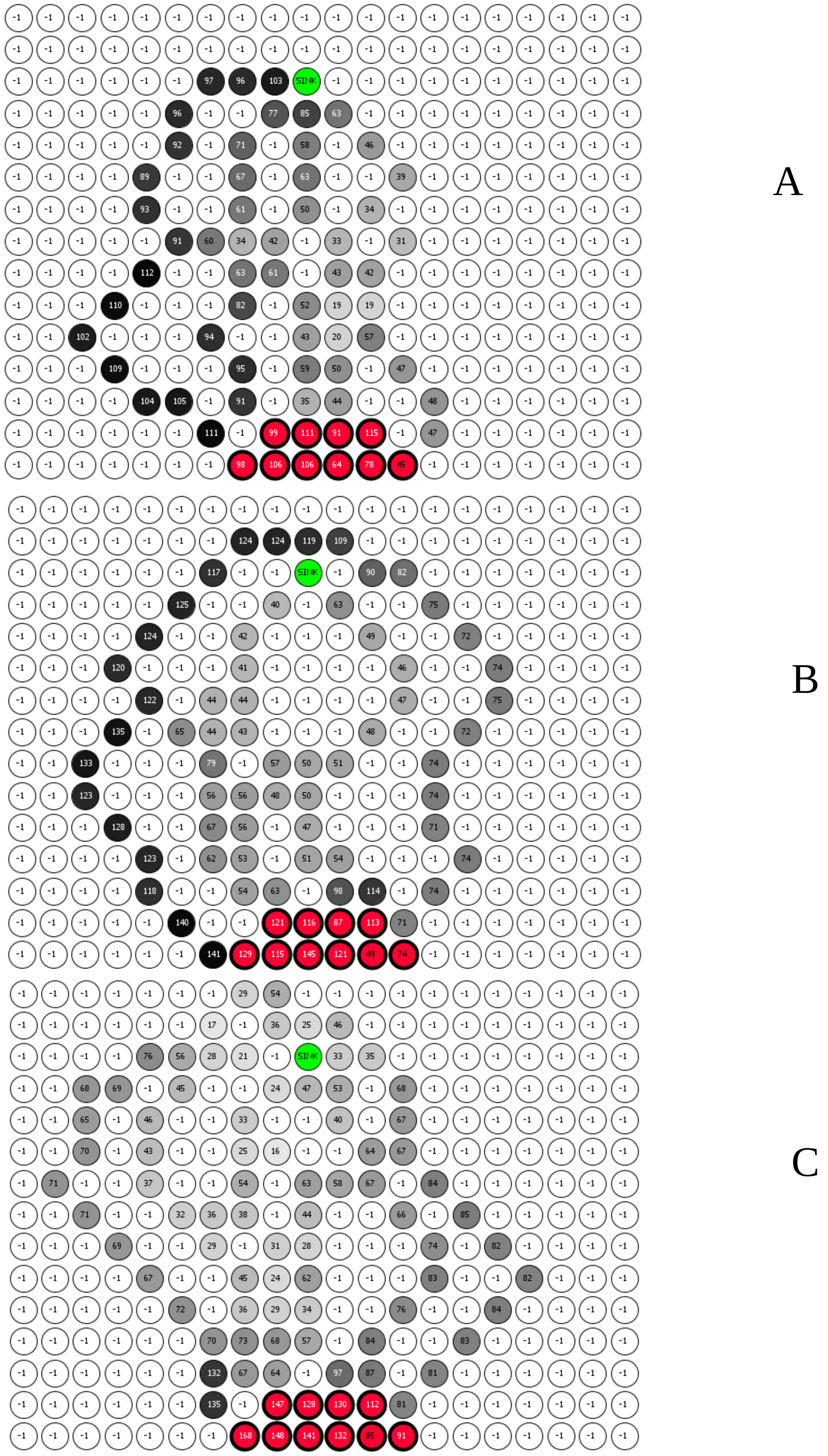
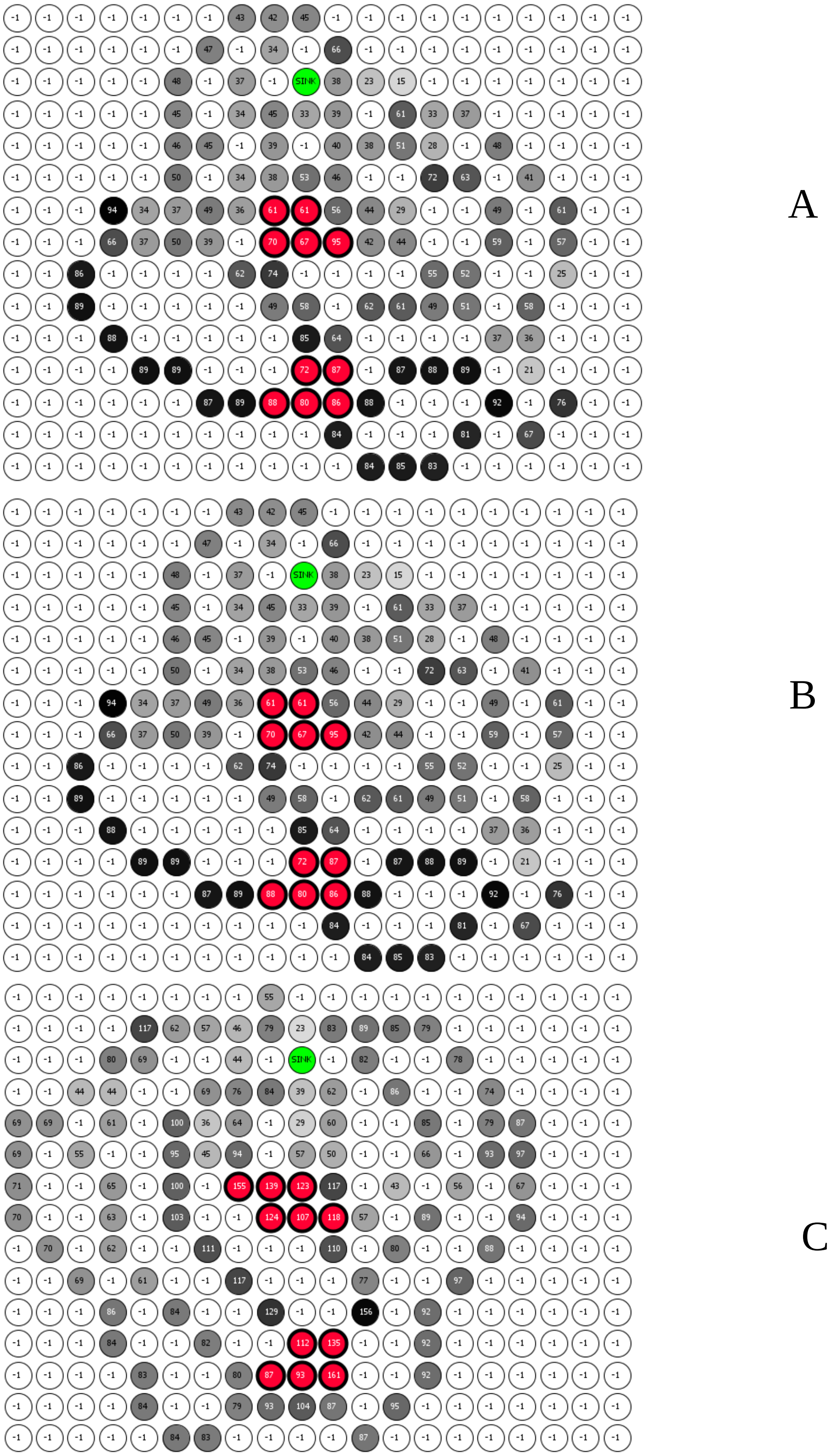


Figure 9: Visual representation of the number of packets arrived to each nodes for scenario 1 at T=60, for (a) CBR=25 pkt/sec (b) CBR=35 pkt/sec (c) CBR=45 pkt/sec



A

-1	-1	-1	-1	-1	-1	-1	43	42	45	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	47	-1	34	-1	66	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	48	-1	37	-1	SINK	38	23	15	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	45	-1	34	45	33	39	-1	61	33	37	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	46	45	-1	39	-1	40	38	51	28	-1	48	-1	-1	-1	-1
-1	-1	-1	-1	-1	50	-1	34	38	53	46	-1	-1	72	63	-1	41	-1	-1	-1
-1	-1	-1	94	34	37	49	36	61	61	56	44	29	-1	-1	49	-1	61	-1	-1
-1	-1	-1	66	37	50	39	-1	70	67	95	42	44	-1	-1	59	-1	57	-1	-1
-1	-1	86	-1	-1	-1	-1	62	74	-1	-1	-1	-1	55	52	-1	-1	25	-1	-1
-1	-1	89	-1	-1	-1	-1	-1	49	58	-1	62	61	49	51	-1	58	-1	-1	-1
-1	-1	-1	88	-1	-1	-1	-1	-1	85	64	-1	-1	-1	-1	37	36	-1	-1	-1
-1	-1	-1	-1	89	89	-1	-1	-1	72	87	-1	87	88	89	-1	21	-1	-1	-1
-1	-1	-1	-1	-1	-1	87	89	88	80	86	88	-1	-1	-1	92	-1	76	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	84	-1	-1	-1	81	-1	67	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	84	85	83	-1	-1	-1	-1	-1	-1

B

-1	-1	-1	-1	-1	-1	-1	43	42	45	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	47	-1	34	-1	66	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	48	-1	37	-1	SINK	38	23	15	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	45	-1	34	45	33	39	-1	61	33	37	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	46	45	-1	39	-1	40	38	51	28	-1	48	-1	-1	-1	-1
-1	-1	-1	-1	-1	50	-1	34	38	53	46	-1	-1	72	63	-1	41	-1	-1	-1
-1	-1	-1	94	34	37	49	36	61	61	56	44	29	-1	-1	49	-1	61	-1	-1
-1	-1	-1	66	37	50	39	-1	70	67	95	42	44	-1	-1	59	-1	57	-1	-1
-1	-1	86	-1	-1	-1	-1	62	74	-1	-1	-1	-1	55	52	-1	-1	25	-1	-1
-1	-1	89	-1	-1	-1	-1	-1	49	58	-1	62	61	49	51	-1	58	-1	-1	-1
-1	-1	-1	88	-1	-1	-1	-1	-1	85	64	-1	-1	-1	-1	37	36	-1	-1	-1
-1	-1	-1	-1	89	89	-1	-1	-1	72	87	-1	87	88	89	-1	21	-1	-1	-1
-1	-1	-1	-1	-1	-1	87	89	88	80	86	88	-1	-1	-1	92	-1	76	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	84	-1	-1	-1	81	-1	67	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	84	85	83	-1	-1	-1	-1	-1	-1

C

-1	-1	-1	-1	-1	-1	-1	-1	55	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	117	62	57	46	79	23	83	89	85	79	-1	-1	-1	-1	-1	-1
-1	-1	-1	80	69	-1	-1	44	-1	SINK	-1	82	-1	-1	78	-1	-1	-1	-1	-1
-1	-1	44	44	-1	-1	69	76	84	39	62	-1	86	-1	-1	74	-1	-1	-1	-1
69	69	-1	61	-1	100	36	64	-1	29	60	-1	-1	85	-1	79	87	-1	-1	-1
69	-1	55	-1	-1	95	45	94	-1	57	50	-1	-1	66	-1	93	97	-1	-1	-1
71	-1	-1	65	-1	100	-1	155	139	123	117	-1	43	-1	56	-1	67	-1	-1	-1
70	-1	-1	63	-1	103	-1	-1	124	107	118	57	-1	89	-1	-1	94	-1	-1	-1
-1	70	-1	62	-1	-1	111	-1	-1	-1	110	-1	80	-1	-1	88	-1	-1	-1	-1
-1	-1	69	-1	61	-1	-1	117	-1	-1	-1	77	-1	-1	97	-1	-1	-1	-1	-1
-1	-1	-1	86	-1	84	-1	-1	129	-1	-1	156	-1	92	-1	-1	-1	-1	-1	-1
-1	-1	-1	84	-1	-1	82	-1	-1	112	135	-1	-1	92	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	83	-1	-1	80	87	93	161	-1	-1	92	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	84	-1	-1	79	93	104	87	-1	95	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	84	83	-1	-1	-1	-1	87	-1	-1	-1	-1	-1	-1	-1	-1

Figure 10: Visual representation of the number of packets arrived to each nodes for scenario 2 at T=60, for (a) CBR=25 pkt/sec (b) CBR=35 pkt/sec (c) CBR=45 pkt/sec

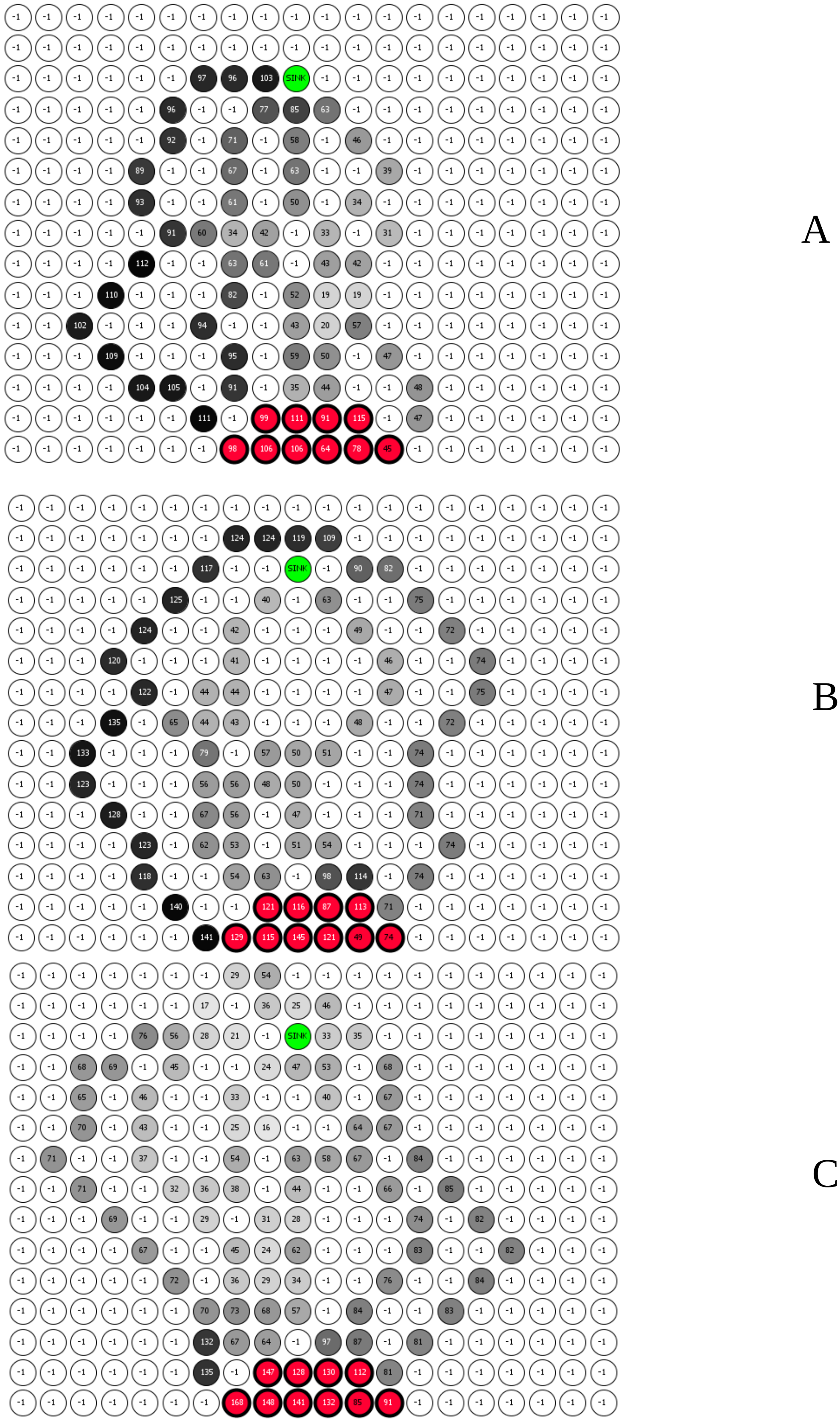


Figure 11: Visual shows the number of packets arrived for scenario 3 at T=30 before failure occurs , for (a) CBR=25 pkt/sec (b) CBR=35 pkt/sec (c) CBR=45 pkt/sec

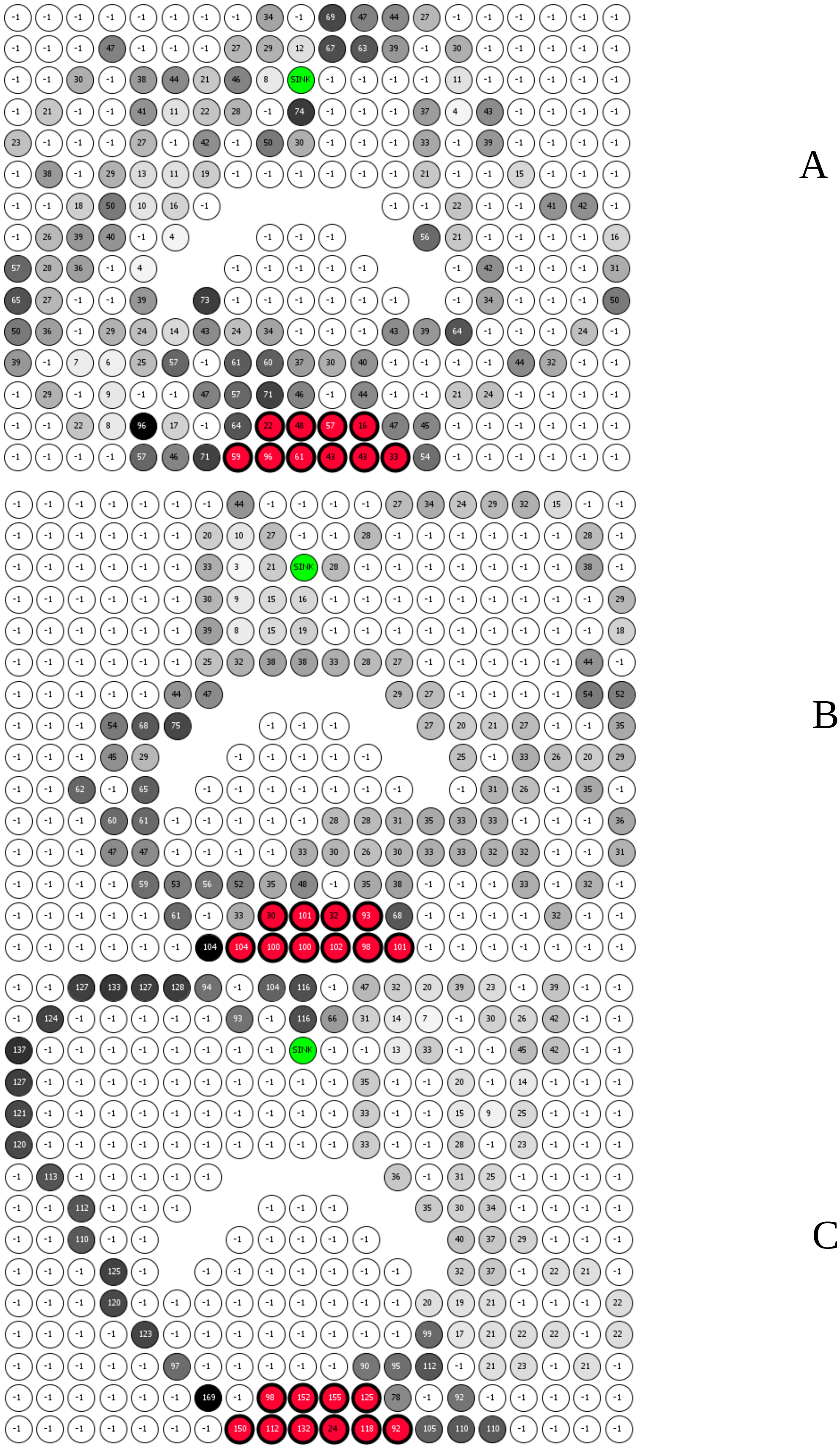


Figure 12: Visual representation of the number of packets arrived to each nodes for scenario 3 at T=80, for (a) CBR=25 pkt/sec (b) CBR=35 pkt/sec (c) CBR=45 pkt/sec

4.6 Parameterization and Graphs

In this section we will project a number of graphs using different parameters for AntCC for comparison with the FlockCC results. Generally we will used 3 scenarios with the same Data Packet Rate which was the 35pkts/sec and for each scenario we extract results during different periods of time (T= 0.5 , 1 , 1.5 , 2) and using the best possible parameter metrics (ξ for FlockCC and a,g,b1,b2,b3 for AntCC). We have tested 12 scenario's and created 4 graphs for comparison which are illustrated below. First shows the Packet Delivery Ration, second shows End-to-End Delay, third shows Overflows and the last illustrates the Collisions. Later in chapter 4.7 we will give a comparison and conclusion disruption about the results overview.

Scenarios 1,2,3 Data Packet Delivery Ratio:

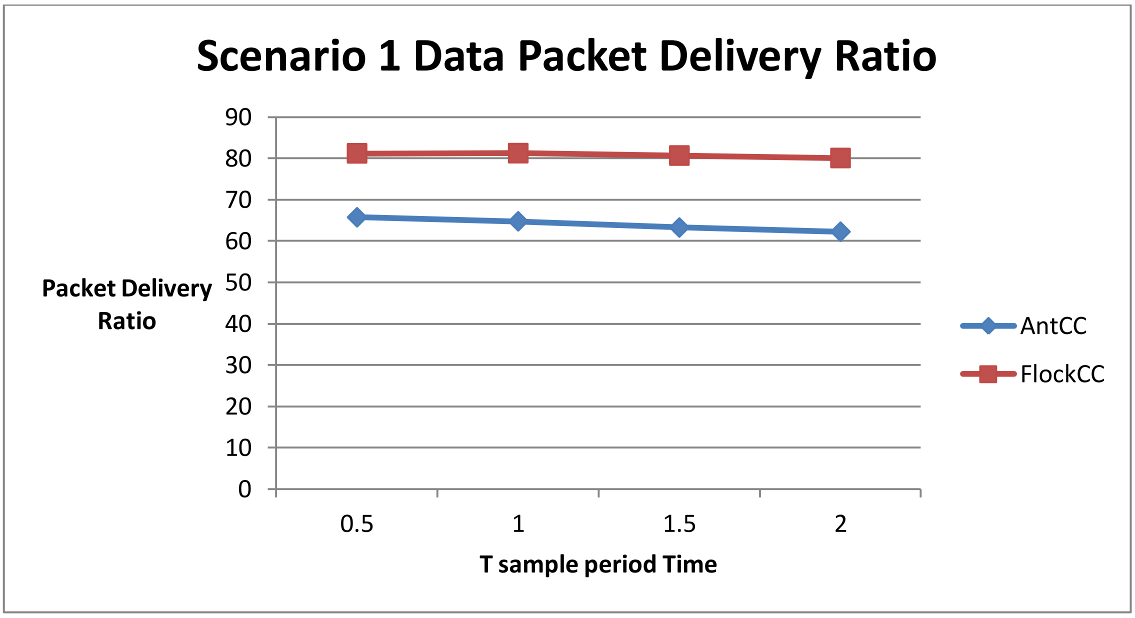


Figure 13: Graphs for Scenario 1 Data Packet Delivery Ratio with 35pkts/sec Data packet rate

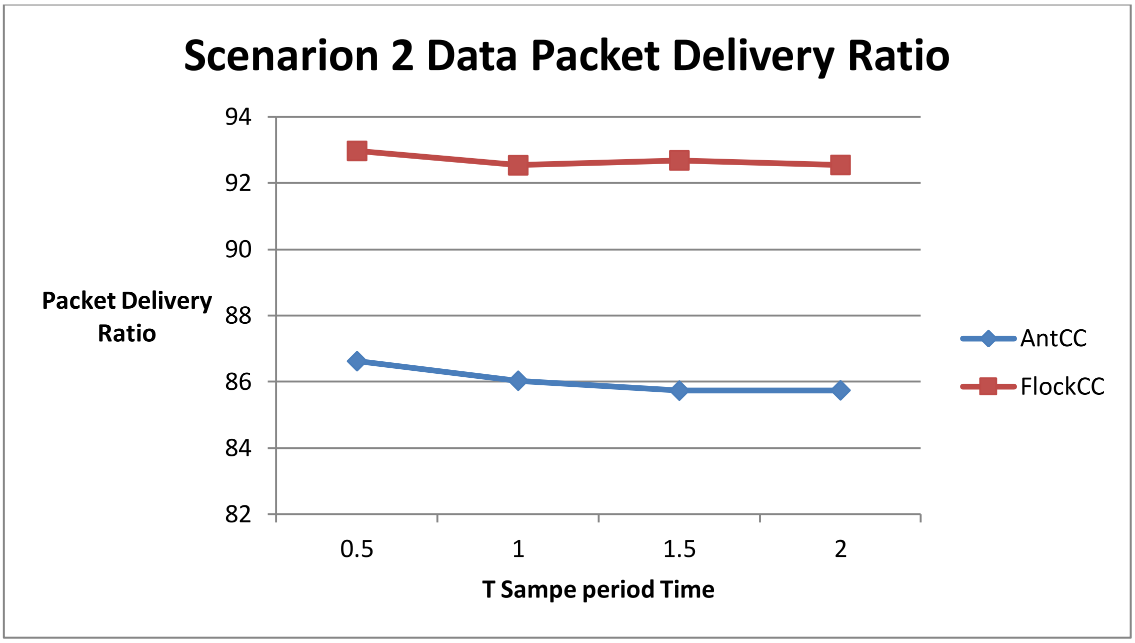


Figure 14: Graphs for Scenario 2 Data Packet Delivery Ratio with 35pkts/sec Data packet rate

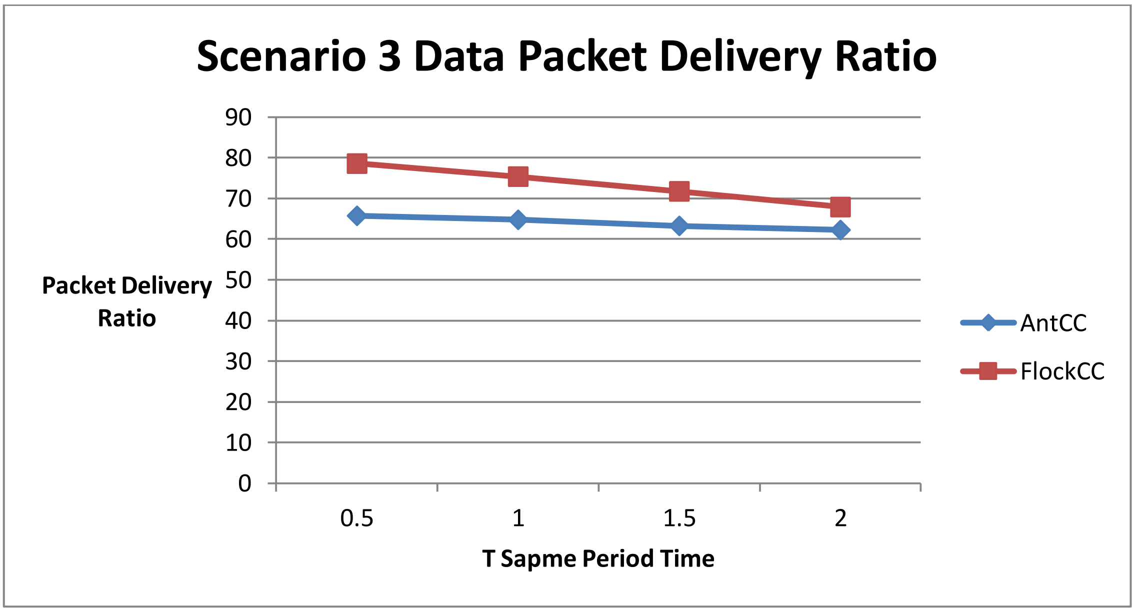


Figure 15: Graphs for Scenario 3 Data Packet Delivery Ratio with 35pkts/sec Data packet rate

Scenarios 1,2,3 End-to-End Delay:

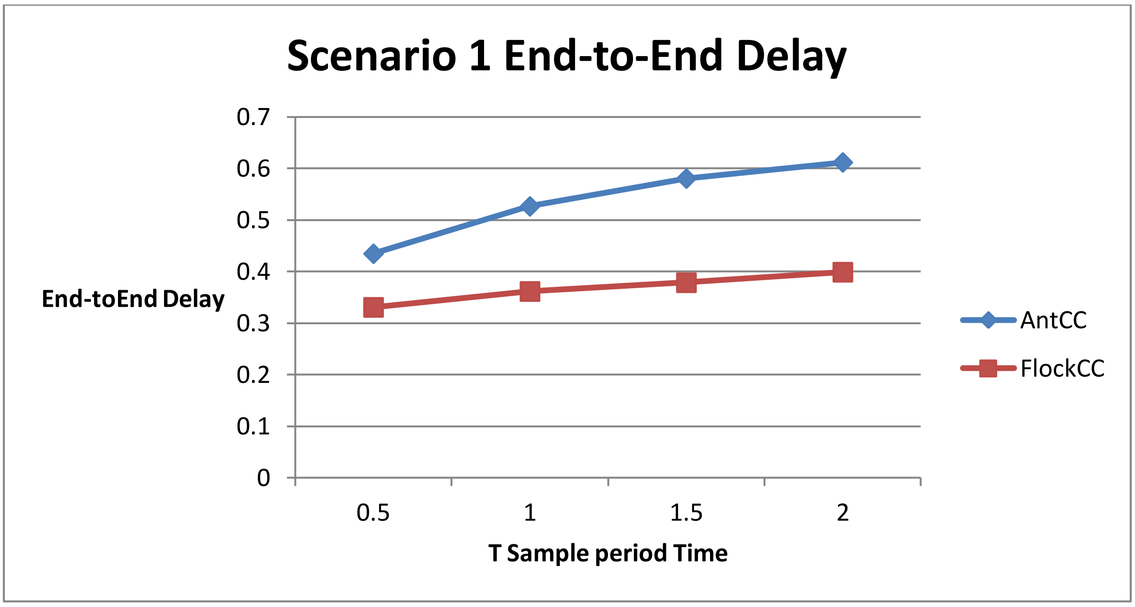


Figure 16: Graphs for Scenario 1 End to End Delay with 35pkts/sec Data packet rate

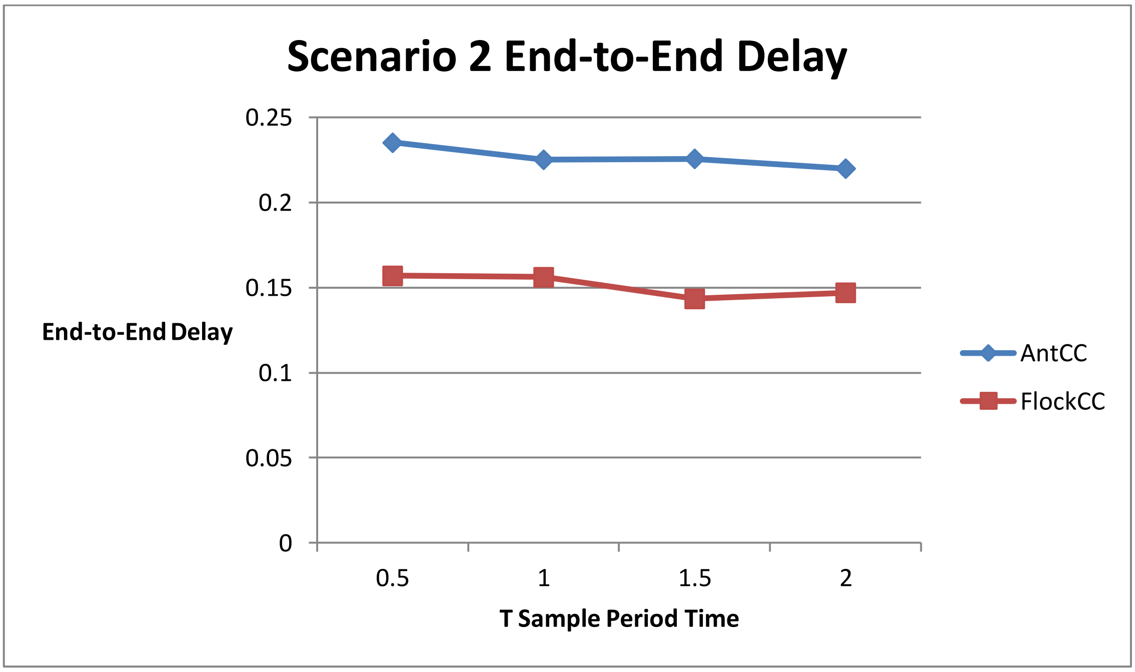


Figure 17: Graphs for Scenario 2 End to End Delay with 35pkts/sec Data packet rate

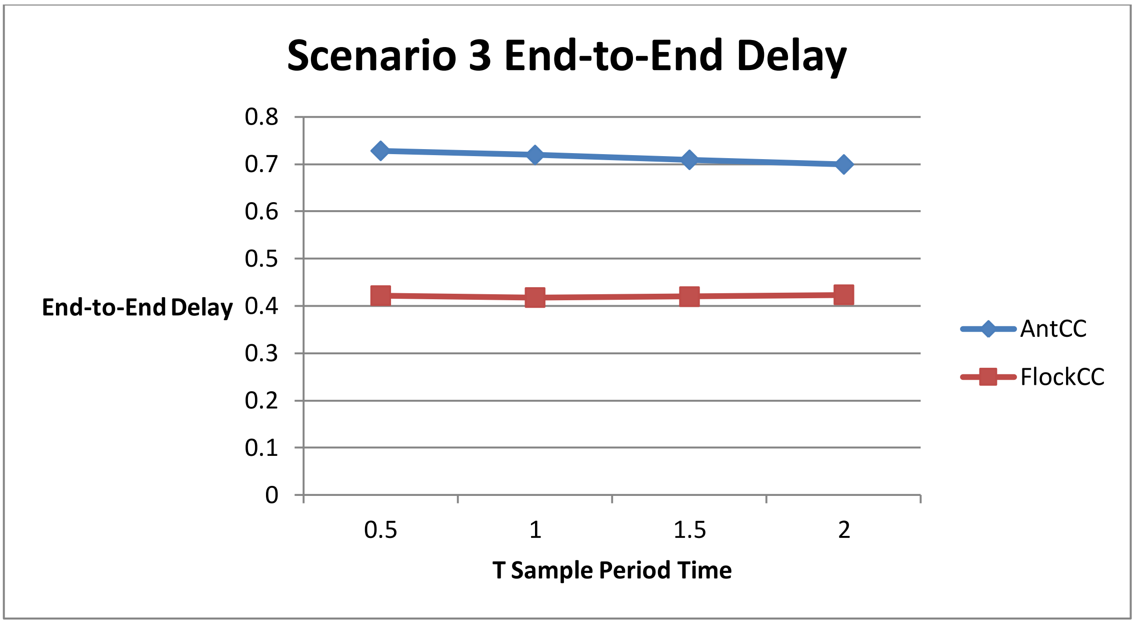


Figure 18: Graphs for Scenario 3 End to End Delay with 35pkts/sec Data packet rate

Scenarios 1,2,3 Overflows:

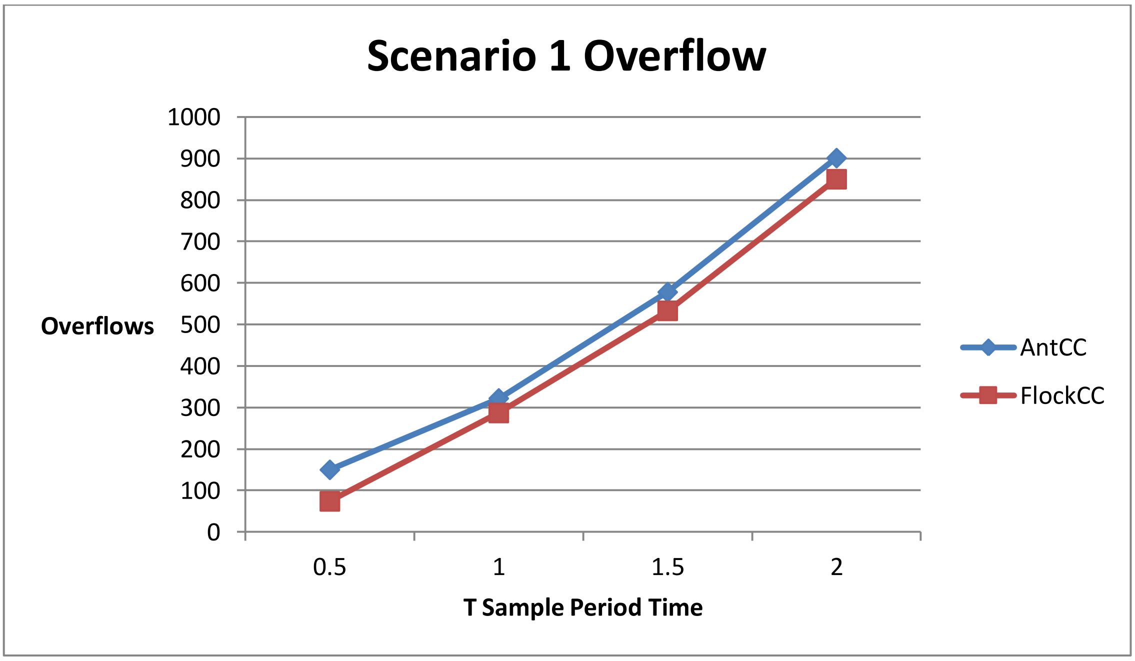


Figure 19: Graphs for Scenario 1 Overflows with 35pkts/sec Data packet rate

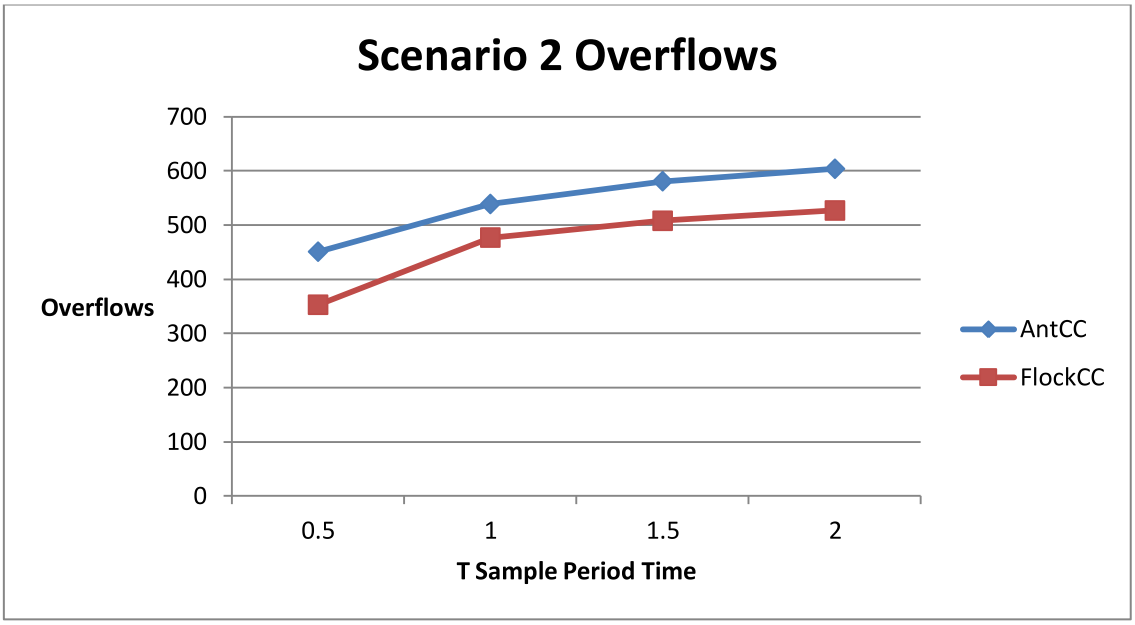


Figure 20: Graphs for Scenario 2 Overflows with 35pkts/sec Data packet rate

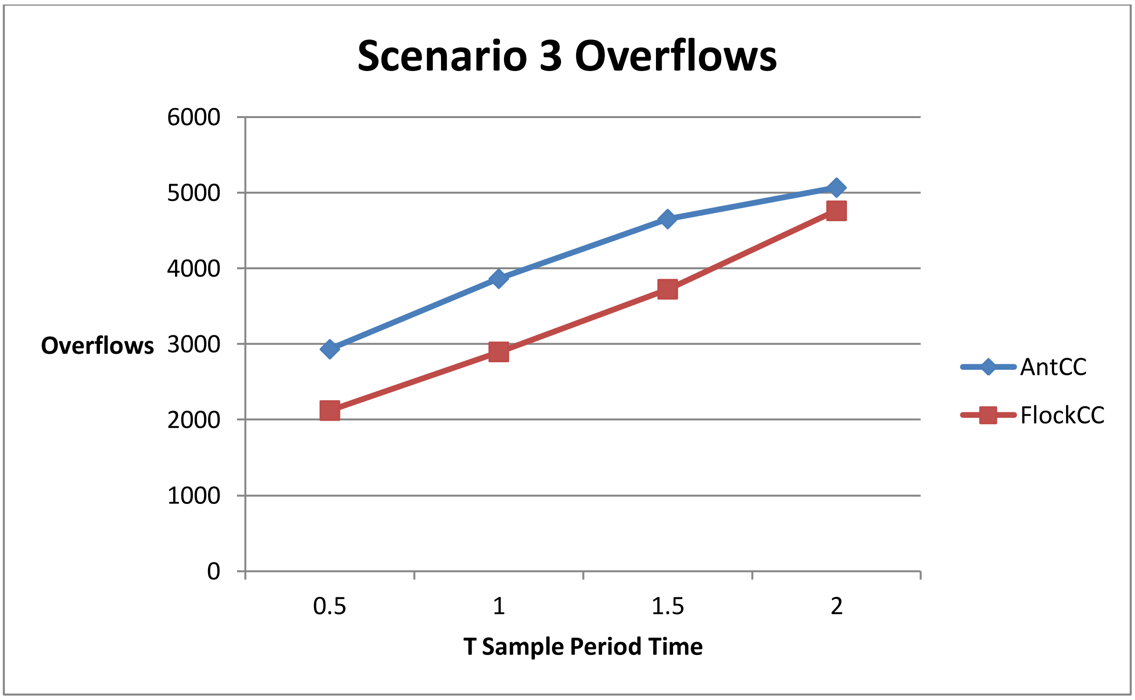


Figure 21: Graphs for Scenario 3 Overflows with 35pkts/sec Data packet rate

Scenarios 1,2,3 Collisions:

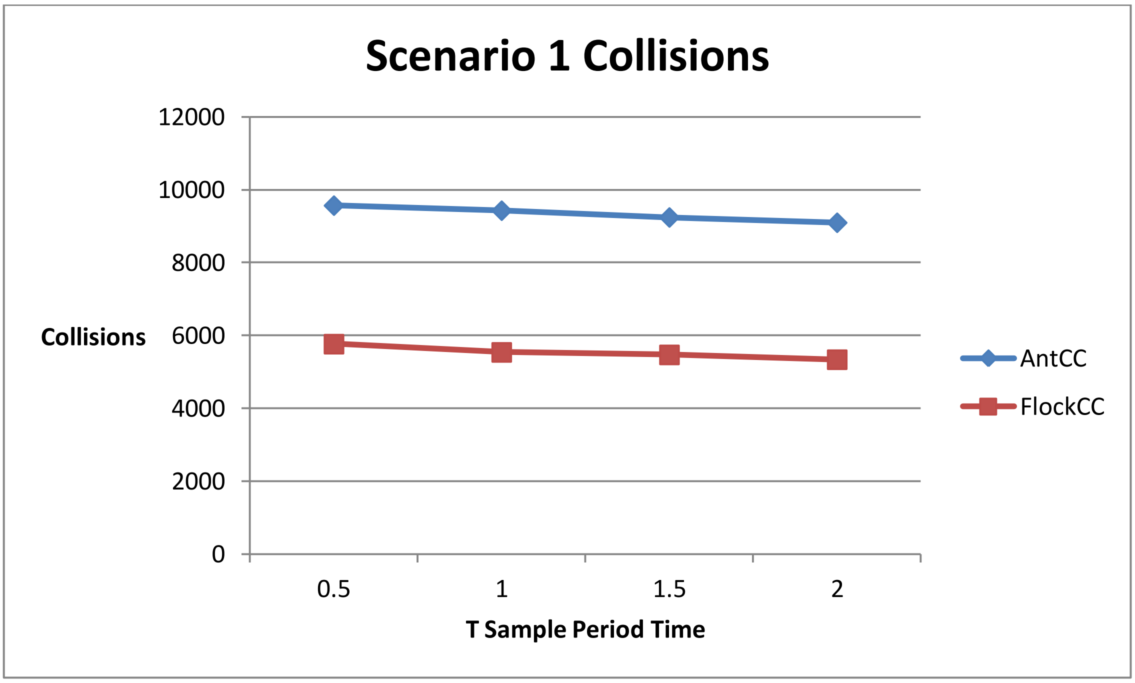


Figure 22: Graphs for Scenario 1 Collisions with 35pkts/sec Data packet rate

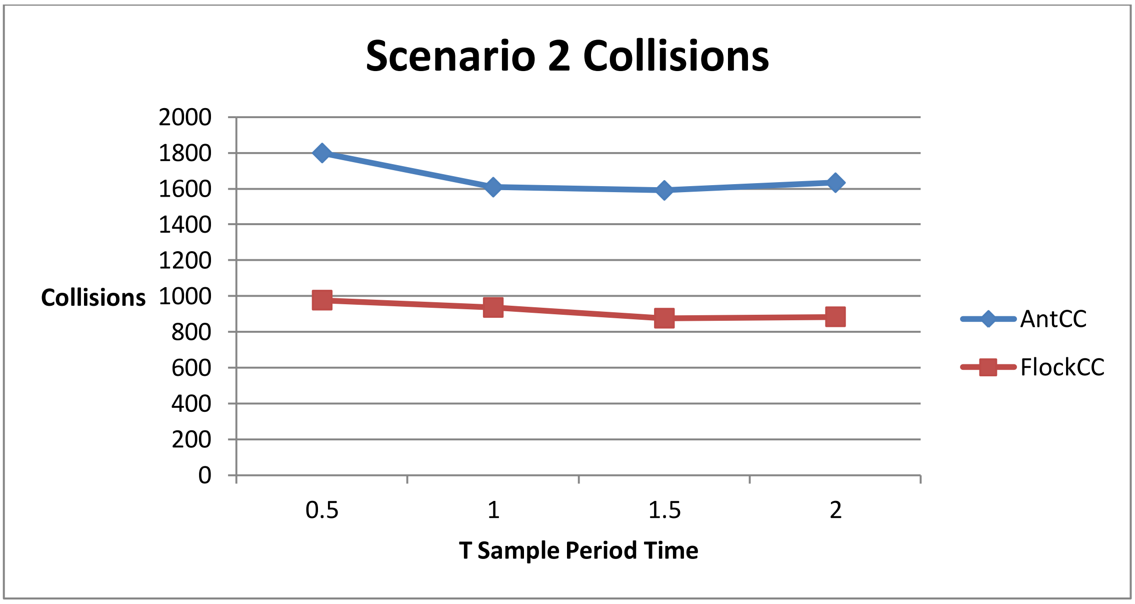


Figure 23: Graphs for Scenario 2 Collisions with 35pkts/sec Data packet rate

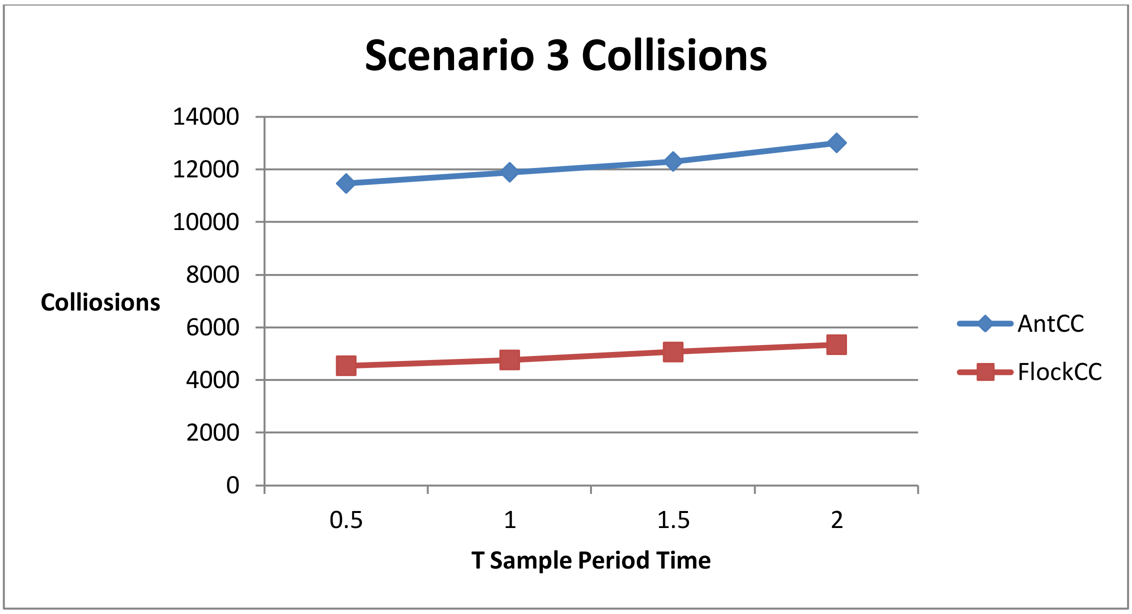


Figure 24: Graphs for Scenario 3 Collisions with 35pkts/sec Data packet rate

4.7 Comparing AntCC and FlockCC Graphs results

In this section we present the results of comparison between the AntHocNet and the FlockCC [3] protocol. Both protocol have been tested for all scenarios and the different parameters. For the comparison of these two protocols we took results and we compare the Data Packet Delivery Ratio which is the total number of packet arrived at the sink , also compare the End-to-End Delay of the packet until they reached the sink, number of total Overflows in all the scenario and last the total number of Collisions in all the scenario case. The Data Packet Rate is keep always in 35 pkts/sec which is consider a congested scenario.

From the figures above for Scenario 1,2,3 for Data Packet Delivery Ratio we can clearly see that FlockCC achieves far better performance than the AntCC. Not even on normal network which is slightly congested but also in cases of Link failure. Also we can see that End-to-End Delay is higher for AntCC rather than FlockCC. A possible reason for that is that AntCC and Ant Colony algorithm uses always the paths with high pheromone value. Ant will search for new path only when the current path not exist anymore. FlockCC is using an algorithm to balance the data packets in the network and when a path is congested data packets are forwarded to nodes which are not congested and have a path for the sink.

AntCC limitation is also cause of a specific aspect of the algorithm which is dropping the reactive forward control packets arriving at the sink node. Sink will keep and send back only one backward and from the received reactive forward ants. This drawback will created only few routing paths for the sender nodes. A results of that in a link failure situation packets cannot find routes available to the destination which will cause the algorithm to enter in Reactive Route Setup Process in order to find paths for the data packets which causes also the End-to-End Delay.

The limited routes for the destination is the basic reason causing Overflows in the routers queues. In our scenarios we keep the routes queue on 50 packets. On data packet rate 35 pkts/sec our network utilization falls immediately and the during the few routes creation from the Reactive Route Setup Process is the outcome of the Overflows in the network.

From the Graphs above we can see the huge amount of difference in the AntCC and FlockCC protocols. The limited data path is also the reason of having more collision in our network. All nodes are trying to send their packets to the node with the highest pheromone value from the equation descript in Chapter 3.3 and this is the outcome of seeing many collisions and Overflows on specific nodes.

Chapter 5

Conclusion

5.1 Conclusion

In this thesis I had to complete and correct the AntHocNet routing protocol in the context of WSNs. This implementation took place at NS-2 network simulator and during the simulation of some scenarios I took some results in case to compare it with the FlockCC routing algorithm. The parameters to compare the two routing protocols are End-To-End delay, Packet Delivery Ratio Overflow and Collision. The results clearly shows that the FlockCC protocol is superior, more robust, efficient and achieves better performance than AntCC protocol. The basic reasons for these are discussed above on chapter 4.7. Generally AntCC protocol is an implementation for Mobile Ad Hoc Networks where most nodes are mobile in the network and as a fact of that the updates are more often, this is the reason why it has the proactive part of the protocols which is lowering the performance of the AntHocNet network. The Pheromone diffusion and the Proactive route maintenance are functions which rapidity and constantly running to maintain and update routes. This aspect is also unless in the context of the WSN.

5.2 Future Work

More work is to be done before the full merits on the AntHocNet protocol in the context of WSNs is to be established.

More future Work that can be done is to add more parameters in case to compare more protocols with different characteristics like Route Discovery time, Failure detection time and also network throughput means even more compare metrics for the protocols.

Also the Visualization of the nodes is done using sketch program in JAVA which gives a snapshot of the scenario for a specific time. We can modified that to run as a video constantly.

Also create more scenario and even more bigger to compare more parameters plus more random topologies and not sorted like now.

New Network Simulator is available now which can give better results and its less consumptive than NS-2. All of our routing protocols should be rewritten to the new Network Simulator.

Finally, comparison with other routing protocols both traditional and novel would add to the understanding of the strengths and shortcomings of this protocol relative to the already existing ones which also need to be implemented at WSNs context.

A new implementation of the algorithm where we eliminate the most proactive and diffusion updates function seems to be more efficient for the WSN context.

Bibliography

[1] Akyildiz Ian, Su Weilian, Sankarasubramaniam Yogesh, and Cayirci Erdal Georgia Institute of Technology, “A survey on Sensor Networks”, IEEE Communications magazine, Vol. 40(8), pp 102-114, August 2002.

[2] Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey”, Computer Networks Journal, Elsevier Science, Vol. 38(4), pp 393–422, March 2002.

[3] Antoniou Pavlos, Pitsillides Andreas, Andries Engelbrecht, Blackwell Tim, “Mimicking the Bird Flocking Behavior for Controlling Congestion in Sensor Networks”. 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010), Invited Paper, Rome, Italy, November 7-10, 2010.

[4] Antoniou Pavlos, Pitsillides Andreas, Blackwell Tim, Engelbrecht Andries, Michael Loizos, “Congestion Control in Wireless Sensor Networks based on the Bird Flocking Behavior”, IFIP 4th International Workshop on Self-Organizing Systems(IWSOS 2009), Zyrich, Switzerland, December 9-11, 2009, pp. 200-205.

[5] Chonggang Wang and Kazem Sohraby, University of Arkansas Bo Li, The Hong Kong University of Science and Technology, Mahmoud Daneshmand, AT&T Labs Research Yueming Hu, South China Agricultural University, “A Survey of Transport Protocols for Wireless Sensor Networks”, IEEE Network, Vol. 20, May/June 2006, pp. 34-40.

[6] Chonggang Wang and Kazem Sohraby, University of Arkansas Bo Li, The Hong Kong University of Science and Technology,Weiwen Tang, Sichuan Communication Research Planning & Designing Co., Ltd., Chengdu, China, “Issues of Transport Control Protocols for Wireless Sensor Networks”. In

Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on, volume 1, pages 422–426, Arkansas Univ., Fayetteville, AR, USA, May 2005.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: Ascalable and robust communication paradigm for sensor networks. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networks (Mobicom), pages 56-67, Boston, MA, USA, 2000.

[8] Ducatelle, F., Adaptive Routing in Ad Hoc Wireless Multi-hop Networks, PhD thesis, Universita della Svizzera Italiana, Istituto Dalle Molle di Studi sull/Intelligenza Artificiale, 2007.

[9] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks: Shortest path is not enough. In Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I). Princeton, New Jersey: ACM SIGCOMM, October 2002.

[10] E. Bonabeau, M. Dorigo, and G. Theraulaz, Inspiration for optimization from social insect behavior, Nature, Vol. 406, 2000, pp. 39-42.

[11] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications.6(2):46–55, April 1999. Proceedings of IEEE ICC'98, pages 156–160, August 1998. [7] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[12] G. Di Caro, F. Ducatelle and L. M. Gambardella, AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks, European Trans. On Telecommunications, Self-organization in Mobile Networking, 16, 2005, pp. 443-455.

[13] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. Computer Networks Journal, Vol. 47:4, pp 445-487, March 2005.

[14] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparisonof Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Proceedings of the Fourth Annual ACM/IEEEInternational Conference on Mobile Computing and Networking, pages 85-97. ACM/IEEE, October 1998.

[15] J. Liu, L. F. Perrone, Y. Yuan, and D. Nicol. The simulator for wireless ad hoc networks (SWAN). Bucknell University, 2005. Available from: http://www.eg.bucknell.edu/swan.

[16] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for distributed discrete optimization. Artificial Life, 5(2), pp. 137-172, 1999.

[17] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76(12), pp. 579-581, 1989.

[18] W. C. Y. Lee. Mobile Communications Engineering: Theory and Applications. McGraw Hill Professional, 1997.

[19] UC Berkeley, LBL, USC/ISI, and Xerox PARC. The ns Manual. Available from: http://www.isi.edu/nsnam/ns/ns-documentation.html.

[20] OPNET Technologies, Inc. OPNET Users' Manual. Avilable from: http://www.opnet.com.

[21] Scalable Network Technologies, Inc. QualNet Simulator, Version 3.8, 2005. Available from: http://www.scalable-networks.com.

[22] http://www.stafflogic.eu/faq/ant-colony-optimization/

[23] http://en.wikipedia.org/wiki/Network_congestion

[24] Sandy A-Mixail ,Implementation and Evaluation of the Biologically -inspired AntHocNet routing protocol in sensor network, May 2011

[25] http://en.wikipedia.org/wiki/List_of_ad_hoc_routing_protocols

[26] http://en.wikipedia.org/wiki/Physical_layer

[27] http://en.wikipedia.org/wiki/Link_layer

[28] http://en.wikipedia.org/wiki/Network_layer

[29] http://en.wikipedia.org/wiki/Transport_layer

[30] Frederick Ducatelle, Adaptive Routing in Ad Hoc Wireless Multi- hop Networks, Lugano, Switzerland, May 2007