2013

Routing, Localization And Positioning Protocols For Wireless Sensor And Actor Networks

Mustafa Akbas
University of Central Florida

Routing, Localization and Positioning Protocols for Wireless Sensor and Actor Networks

by

Mustafa İlhan Akbaş
B.S. Electrical and Electronics Engineering, Middle East Technical University, 2003
M.S. Electrical and Electronics Engineering, Middle East Technical University, 2006

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2013

Major Professor:
Damla Turgut

ABSTRACT

Wireless sensor and actor networks (WSANs) are distributed systems of sensor nodes and actors that are interconnected over the wireless medium. Sensor nodes collect information about the physical world and transmit the data to actors by using one-hop or multi-hop communications. Actors collect information from the sensor nodes, process the information, take decisions and react to the events.

This dissertation presents contributions to the methods of routing, localization and positioning in WSANs for practical applications. We first propose a routing protocol with service differentiation for WSANs with stationary nodes. In this setting, we also adapt a sports ranking algorithm to dynamically prioritize the events in the environment depending on the collected data. We extend this routing protocol for an application, in which sensor nodes float in a river to gather observations and actors are deployed at accessible points on the coastline. We develop a method with locally acting adaptive overlay network formation to organize the network with actor areas and to collect data by using locality-preserving communication. We also present a multi-hop localization approach for enriching the information collected from the river with the estimated locations of mobile sensor nodes without using positioning adapters. As an extension to this application, we model the movements of sensor nodes by a subsurface meandering current mobility model with random surface motion. Then we adapt the introduced routing and network organization methods to model a complete primate monitoring system. A novel spatial cut-off preferential attachment model and

center of mass concept are developed according to the characteristics of the primate groups. We also present a role determination algorithm for primates, which uses the collection of spatial-temporal relationships. We apply a similar approach to human social networks to tackle the problem of automatic generation and organization of social networks by analyzing and assessing interaction data. The introduced routing and localization protocols in this dissertation are also extended with a novel three dimensional actor positioning strategy inspired by the molecular geometry. Extensive simulations are conducted in OPNET simulation tool for the performance evaluation of the proposed protocols.

ACKNOWLEDGMENTS

I would like to thank OPNET Technologies Inc. for supporting my research by providing OPNET Modeler software under OPNET University Program.

I would like to thank my advisor, Dr. Damla Turgut who provided me the vision to proceed and gave me the freedom to explore through every step of my Ph.D study. I appreciate Dr. Michael Georgiopoulos, Dr. Mostafa A. Bassiouni, Dr. Ladislau Bölöni and Dr. Yue Zhao for serving as members of my dissertation committee. Their suggestions and advices have helped me immensely to improve my research. I would like to thank Dr. Matthias R. Brust for creating time to meet me face-to-face or online whenever I needed.

My sincere thanks go to my mentor at my first job, Semih Gül, and my M.S. advisor Dr. Cüneyt Bazlamaçı, both of whom encouraged me to pursue a Ph.D. I owe an appreciation to my friends Onur Gülçat, Sertaç Cinel, Doğa Somer and Dr. Doğa Gürsoy, who have been the sources of laughter and support for over twenty years and never let me feel alone.

I thank my fellow labmates, Drs. Swastik Brahma, Yi Luo, Wenjing Wang and Gürkan Solmaz for their assistance and support over the years. I was lucky to meet Dr. Zubair Ahmad as my first labmate. I have had his support both in research and in daily life throughout my Ph.D. I would like to thank Dr. Tonguç Öztek, Fırat Şeker, Atilla Cingöz and Dr. Mustafa Gül for helping me in any sorts of issues ranging from my scientific questions to fixing my car. Special thanks to my long lost sister Melike Minareci, positive energy booster Henriette Schoen and the best cook in Orlando, Dr. Maria Bull.

I am forever grateful to my parents, Songül and Hilmi Akbaş, who always provide unflagging support. It wouldn't be even possible to start and continue this work without feeling their endless love. My older brother, Dr. Sami Oğuzhan Akbaş, has been an inspiring example in my life; I thank him, my sister-in-law Selvet Duygu and my nephew, Batuhan. I also want to thank my family-in-law İsfendiyar, Arzu and Esin Söyler for their support and the joy they brought into my life.

I feel fortunate to have the chance for opening a new chapter of my life in Orlando. I not only worked with great people, but also met my wonderful wife, Aslı Söyler Akbaş. Her encouragement, patience, love and ingenious ideas have been my invaluable resources to finish this work. I can only "promise her anything" to express my gratitude.

Finally, this dissertation is dedicated to my grandmother, Hatice Akbaş, whom I lost during the time of my Ph.D study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

This dissertation presents contributions to the fields of routing, network organization, localization and actor positioning in wireless sensor and actor networks (WSANs) [1,2]. The first routing solution presented in this dissertation takes the properties of distinct node types as a driving factor and provides service differentiation. This routing solution is also extended with a ranking method to compare the performances of different types of traffic in terms of the changes in observed values. The support for efficient resource utilization is a critical part of WSAN routing protocols because of the limited node resources and the coexistence of different node types. We then extend this routing protocol for WSANs with mobile sensor nodes by introducing a locally acting adaptive overlay network formation. For the same application scenario, a localization method is designed to augment the collected information about the physical world by adding data to be used in the localization of the events. A realistic current mobility model is also adapted according to the requirements of the scenario to analyze the accuracy of the localization algorithm. The network organization strategies for routing and localization protocols focus on the topological requirements of the application scenarios. We adapt these strategies to form the basis for an animal monitoring application. In this application, the information gathered by the network organization is used to interpret the social structure of a primate community. Then the approach used in primate monitoring system is adapted for human social networks for friend ranking and social network map generation. In the literature, the routing and localization protocols are improved by using node positioning strategies and most of the approaches are for two dimensional space. We consider the problem of actor positioning in three dimensional space and apply the theory of valence shell electron pair repulsion (VSEPR) theory for the solution.

The remainder of this chapter is organized as follows. We provide a brief introduction to WSANs in Section 1.1. We motivate the development of novel routing, localization and positioning methods in Section 1.2. Finally we present a summary of our contributions in this research domain in Section 1.3.

1.1 Wireless sensor and actor networks

WSANs are distributed systems of sensor nodes and actors that are interconnected over the wireless medium. Sensor nodes are small devices with limited data processing capabilities, low transmission rates, small batteries, and short memories. Sensor nodes collect information about the physical world and transmit the data to actors by using one-hop or multi-hop communications. Actors on the other hand have better computation and communication capabilities, larger memories and longer lasting batteries compared to the sensor nodes. WSANs may also have sinks, which act as the leaders of the actors and take decisions about the functions of the network.

In literature, there are different terms used for the nodes in a sensor network, which have better capabilities than the sensor nodes. "Actuator" is a conventional term used for this type of nodes. However an actuator is generally defined as a device to convert an electrical control signal to a physical action, and constitutes the mechanism by which a node acts in the environment [3]. On the other hand, an actor can use one actuator or several actuators to act on the environment. Additionally, an actor not only acts on the environment but also performs networking related functionalities such as receiving, transmitting and processing data. Hence, a panning camera collecting information from the environment and processing the information to decide on the turning direction is an actor while the motor in the camera can be considered as an actuator. The term "actor" is used in this dissertation since the considered application scenarios include data collectors, which process the collected information and use the results for decisions related to the presented protocols.

WSANs can be employed for various applications such as intelligent transportation, environmental monitoring, battlefield operations, healthcare monitoring, animal control and so on. Conventional wireless sensor network (WSN) [4] applications are generally limited to observation of the physical world. The heterogeneous structure of node types and resources allows WSANs to deal with a wider range of possible applications.

1.2 Motivation

The heterogeneous node structure is one of the most important motivations when designing the routing solutions in WSANs. The main characteristics of the communication in WSANs are based on sensor-sensor, sensor-actor and actor-actor coordinations. The limited resources of sensor nodes is already a critical constraint [5] as it is in WSNs, whereas actors generally form an overlay network with stronger resources. Therefore communication solutions applied in traditional wired networks or the solutions in WSNs are generally not suitable for WSANs. The main goals of our routing protocols are to provide coordination among nodes and have the actors handle the computationally expensive tasks.

The development of a method for sensor nodes to affiliate with the actors forms an important part of any WSAN application. We use clustering in network organization to form actor areas. The clustering methods use various parameters in the affiliation process such as distance, energy level, capabilities of the nodes and so on. The specifications of the application scenario and the node properties are important parameters to choose the most appropriate clustering protocol. There are generally multiple sensor nodes affiliated with an actor. In some cases, such as in our localization approach, a single sensor node can be affiliated with multiple actors. Another defining factor for the clustering methods is the transmission from sensor nodes to actors, which can be restricted to a single hop or multi hop communications.

The quality of service (QoS) support is a vital part of various WSAN applications [6] to differentiate the resource usage in the network for different types of events. Similar to communication and coordination approaches, WSANs cannot be simply regarded as WSNs when aiming for QoS support in routing protocols because of the varying degree of node resources. The collected information from the environment depends on the requirements during the system deployment. The priority of different types of information depends on the requirements of the system. Let us consider a fire alarm system. Information about a sharp temperature increase is high priority and must be transmitted by all the nodes in the system without regards to the resources. Periodic reporting of normal temperatures on the other hand is less critical. The network organization and routing strategies can also be combined to offer service differentiation.

We use the term *interest*, popularized by the Directed Diffusion model [7], to define the types of events in the system. In Directed Diffusion, the sink expresses a set of interests regarding the information to be collected. In most of WSAN applications, the interests and their requirements are predefined and remain at their initial values during the lifetime of the network. However, the importance of the events observed by sensor nodes in response to interests may change with the changing conditions of the network. The critical events for most of the applications are the ones with fluctuating or changing observed values rather than the events with constant values. Therefore the priorities of interests must be adjusted according to the changes in the observed values in the network.

In WSANs with mobile nodes, most applications require location information to be associated with the collected data [8]. The sensor nodes can gather various kinds of location-related information when equipped with appropriate measurement technologies [9]. The distance between two nodes, the network connectivity and the strength or angle of signal arrival are widely used to support the estimation of the positions of nodes in the environment. The localization methods generally use *a priori* information about the environment such as the positions of some specific nodes or possible positions that the nodes can be located in the environment [10]. The use of a realistic mobility model is critical to analyze the accuracy of the localization algorithms designed for WSANs with mobile sensor nodes.

We adapted the routing and network organization methods used for WSANs with mobile nodes for an application of monitoring a group of animals with complex social structure such as primates. Conventional approaches for wildlife monitoring require technically sophisticated processes to overcome various complex issues such as the highly invasive structure of most methods, the choice of correct sampling method or the effort to plan and execute a monitoring project [11]. The latest advances in WSANs can be used to overcome many of these drawbacks and challenges. Thus the monitoring of animal groups has been an application area in WSNs and WSANs. However, the research has been mostly on animals with simple social structures, whereas animals with more complex social dynamics such as primates is still an open research area in WSANs. Additionally, the available real life mobility data for most of the animals with complex societies is very limited. Consequently, it is also crucial to design a proper mobility model derived from the expected mobility patterns of the animals under observation.

The routing and localization strategies presented in this dissertation can be improved by efficient actor positioning. Most of the literature on dynamic node positioning strategies is limited to two dimensional space. In recent years, there has been an increasing interest in applications of sensor networks in three dimensional (3D) space such as space exploration, airborne and underwater surveillance, oceanic studies, and storm tracking. The well-known strategies designed for two dimensions become NP-hard in 3D space. The optimization strategies for 3D node positioning are important for efficient data collection. The existing solutions are only preliminary and do not take into consideration the requirements of specific scenarios. Therefore, simple yet effective strategies for 3D node positioning are needed in WSANs.

1.3 Contributions

Our work is to design, implement, and evaluate protocols for routing, localization and positioning in WSANs. More specifically, the major contributions of this dissertation are as follows:

- *Routing and network organization.* We present a lightweight routing protocol with QoS support [12] for stationary WSANs. Our protocol provides QoS by differentiating the rates among different types of interests with dynamic packet tagging at sensor nodes and per flow management at actor nodes. In this setting, we also introduce a method [13] to dynamically determine the weights of the interests by using a nonsensitive ranking algorithm, which depends on the variation in the observed values of collected data. We extend the lightweight routing protocol for WSANs with mobile sensor nodes [14]. Then we integrate it with a network organization protocol [15], in which the actor nodes are pre-assigned cluster heads and multi-hop clusters are formed as the sensor nodes move in the environment.

- *Localization.* We propose a novel approach for the localization of mobile sensor nodes floating in a river [16, 17]. The sensor nodes communicate only with their neighbor nodes. The collected information through this local communication is used to estimate the locations of the events without any GPS receivers at the sensor nodes. To model the movements of the sensor nodes, we adapt a realistic meandering current mobility model [18]. The motion of the sensor nodes in this model follows the advection of the fluid parcels, which is modeled as a combination of a central streamline with a meandering motion around the surface. To the best of our knowledge, this is the first example of using meandering current mobility model with random surface motion for a WSAN operating on the surface of a river.

- *Animal monitoring and role determination.* We provide network formation and mobility models [19] to model the mobility of a group of primates to be used for social life monitoring. A novel spatial cut-off preferential attachment model and a center of mass concept are used and extended for the models. Then we develop a social role determination protocol [20, 21] for capturing and monitoring the social interactions of primates. The nodes are intended to be attached to the primates forming a mobile network. The local interaction patterns among the nodes are analyzed and the roles of animals are determined based on the research on primate social structure. The results of our approach is compared to real-life primate networks using various social network metrics. The role determination approach is adapted to human networks to develop an ego network generation and friend ranking protocol [22]. This protocol generates the social network of a person by using different sources of available interaction data such as physical proximity, text messages, phone calls and video chats. The approach is applied to a real-life dataset of a group of high school students.

- *3D Positioning.* We develop an actor positioning strategy [23–25] for aerial WSANs considering the scenario of toxic plume observation after a volcanic eruption. The positioning algorithm utilizes the Valence Shell Electron Pair Repulsion (VSEPR) theory of chemistry, which is based on the correlation between molecular geometry and the number of atoms in a molecule. We improve the positioning strategy for real-world scenarios by utilizing a rotatable hybrid antenna model (O-BESPAR), which combines the complimentary features of an isotropic omni radio and directional antennas [26]. The characteristics of different antenna modules are analyzed and the rotatable hybrid antenna model is utilized with actor-sink communication, actor rearrangement algorithms, and beamforming.

1.4 Outline

This dissertation is organized as follows.

Chapter 1 presents the problem definitions. Chapter 2 conducts a literature review covering the background knowledge for the remaining chapters.

Chapter 3 introduces the lightweight routing protocol with QoS support (LRP-QS) for WSANs. We also present a variant of Colley ranking method as our dynamic interest ranking protocol for WSANs. Then, we present the results of the simulation study, in which LRP-QS is compared with QoS Based Routing Protocol (QBRP) [27] and the proposed ranking method is compared with win-percentage ranking method.

Chapter 4 describes Self Organized and Fair Routing Protocol (SOFROP), an integrated solution of network organization and routing algorithms. We introduce the Amazon River application scenario and discuss topology-related concepts. The system model is proposed for wireless networks with mobile sensor nodes and stationary actors. We illustrate the performance of SOFROP in a simulation study.

Chapter 5 describes the multi-hop localization protocol (MHOPLA) for the Amazon River scenario. The network organization is described with an emphasis on its differences compared to SOFROP. Then the localization method is presented, followed by the explanation on how the location information is used for path estimation. Additionally, the mobility of the sensor nodes are modeled by adapting a subsurface current mobility model. Finally, we present a series of simulation studies investigating the precision of the proposed methods for the targeted scenario.

Chapter 6 introduces network formation and mobility algorithms to provide a complete model of a primate group for social life monitoring. These methods are based on the center of mass and preferential attachment concepts. In addition, an algorithm for capturing the social structure of an animal group is presented. This algorithm is also extended for human networks. Simulation results show the outputs of the models and the performance of the role determination algorithm for different metrics.

Chapter 7 describes an actor positioning algorithm for aerial WSANs (APAWSAN). The goal of the approach is to improve the on-site monitoring of the plume in a volcanic eruption scenario. The application scenario is investigated as a 3D node positioning problem for an aerial WSAN with a central node. Then the approach is improved by utilizing a rotatable hybrid antenna model. The simulation study analyzes the performance of the protocol in terms of packet reception ratio, network reorganization delay, coverage and cardinality of actor nodes.

Chapter 8 concludes the dissertation while providing few possible extensions for future work.

CHAPTER 2
RELATED WORK

2.1 Network organization and routing

Clustering is employed frequently for network organization in wireless ad hoc networks. The clustering algorithms in traditional sensor networks [28] are often used to create a structure of an otherwise flat network topology [29], [30], [31]. The cluster heads have more energy and computation power compared to the regular nodes in most of these protocols, similar to the actors in WSANs (see [32], [33], [34], [35], [36]). The design of the routing schemes proposed for clustered networks is based on the selection of cluster heads and the network structure. The network organization of our protocol shares several aspects with cluster-based routing schemes. The cluster head selection approach by Soro and Heinzelman [33] favor nodes deployed in densely populated network areas to maintain the full network coverage. Smaragdakis et al. [37] propose stable election protocol (SEP), a heterogeneous-aware protocol to prolong the time interval before the death of the first node in a stationary network. For the election of cluster head, SEP requires the energy levels of all nodes, which is used as the metric for heterogeneity. Aslam et al. [38] finds optimal geographical locations for actor nodes with respect to their associated cluster heads. Zhang et al. [39] also use geographical locations of nodes and select cluster heads in adjacent groups of nodes close to one another. This property of the network is used to reduce the average energy consumption in the WSN. Chen et al. [40] introduce a dynamic clustering algorithm for target tracking, which creates a hierarchical network structure. Clustering Patch Hierarchical Routing Protocol (CPHRP) by Lin and Liao [41] uses network coverage rate and effective network lifetime to evaluate a WSN. The main objective of CPHRP is to optimize network coverage rate through clustering patch by using these metrics and hierarchical multi-path tree routing. The clustering routing protocol by Boukerche et al. [42] aims at optimizing energy dissipation in the network while providing fault tolerance. The algorithm alternates the inter-cluster communication nodes and possible routes to the sink to reduce energy expenditure.

There are examples of clustering algorithms in mobile ad hoc networks [32], [43], [36], in which the cluster heads are permanent. Thus cluster head election procedure is obsolete for these algorithms. However, the network and the clustering algorithms must be designed in such a way that the actor node is always the most attractive cluster head in its surrounding. Furthermore, only a few clustering algorithms allow multi-hop clusters, i.e., clusters where cluster members can potentially be several hops away from the cluster head [44]. Since actor nodes are specially equipped nodes to aggregate and process data while delivering a long life-time, the number of actor nodes must be minimized. This property of WSANs reduces the number of cluster heads required by the network. This is also important when actors cannot be deployed very close to each other due to restricted access to the environment.

Support for efficient resource utilization is an important part of WSAN network organization and routing protocols. Therefore QoS becomes a critical part of the communication protocols used in WSANs [6], [45]. Throughput, delay, jitter, and packet loss are among the most fundamental QoS metrics used to measure the degree of efficiency in these services [46], [40], [47].

The routing paradigm by Hu et al. [48] builds an anycast tree rooted at each event source to reduce latency and energy consumption in communication. The dynamic behaviors of the sinks for joining or leaving the system shape these anycast trees. Cañete et al. [49] present a component-based framework, which combines macro-programming and node-centric programming to develop applications over WSANs with the specification of real-time constraints between services. Tao et al. [50] propose the flow-balanced routing (FBR) protocol, which aims to achieve power efficiency and coverage preservation. FBR assigns the transferred data over multiple paths from the sensor nodes to the sink to equalize the power consumption of nodes. Melodia et al. [51] use an event-driven clustering paradigm to design a sensor-actor coordination model and formulate the actor coordination as a task assignment optimization problem. The real-time routing framework by Shah et al. [52] addresses the coordination of sensor and actor nodes through the delay bound for distributed routing. Another coordination algorithm among actors is introduced with the real-time communication framework by Ngai et al. [53], where an event reporting algorithm for sensor-actor communication is also given to minimize the transmission delay. Ad Hoc On Demand Delay Constrained Distance Vector Routing (AOD^{2}V) by Sama and Akkaya [54] also uses delay as the main constraint. AOD^{2}V uses delay-EDD at admission control and EDF is used to determine the departure order of the packets at the intermediate nodes. The routing algorithm by Hung et al. [55] determines the maximum amount of data each node can transmit by taking energy as the main parameter. The data transmission protocol by Morita et al. [56] thrives to enhance reliability with redundancy. Xia et al. [6] apply feedback control for dynamic bandwidth allocation, which uses deadline miss ratio control to improve QoS in terms of reliability. Paruchi et al. [57] proposed a distributed and randomized communication protocol with a fairness feature regarding power savings of the sensor nodes, which make local decisions on whether to sleep, or be active based on the energy levels of their neighbors. Energy-balanced routing method based on forward-aware factor (FAF-EBRM) by Zhang et al. [58] uses the link weight and forward energy density to select the next-hop node. FAF-EBRM also uses a spontaneous reconstruction mechanism for local topology to balance the energy. In majority of these algorithms, the main objective is the energy efficiency at the sensor nodes. Directed Diffusion (DD) by Intanagonwiwat et al. [7] achieves energy savings by selecting empirically good paths and by data aggregation. The sink in DD expresses a set of interests regarding the information to be collected and each node records the neighboring node from which the interest is received.

QoSNET by Houngbadji and Pierre [59] takes the network lifetime as the main metric and formulates the QoS routing in large scale wireless networks as an optimization problem to extend the network lifetime. InRout [60] by Villaverde et al., a QoS aware route selection algorithm for WSNs, enables distributed route selection by sharing the local information among neighboring nodes and using Q-learning techniques. The management protocol for reactive sensor and actor systems by Baunach [61] focuses on memory and offers a collaborative approach. Grid-based Multipath with Congestion Avoidance Routing protocol

(GMCAR) [62] divides the network into grids and selects a master node in each grid. These master nodes route packets using the grid densities and the hop count. Energy Efficient and QoS aware multipath routing protocol (EQSR) by Ben-Othman and Yahya [63] maximizes the network lifetime by balancing energy consumption across a set of available paths in the network. The metrics used by EQSR to predict the next hop are residual energy, node available buffer size, and Signal-to-Noise Ratio (SNR). EQSR also employs a queuing model to handle real-time and non-real-time traffic through service differentiation. Hammoudeh and Newman [64] present cluster-based Route Optimisation and Load-balancing protocol (ROL) and . an algorithm for load balancing, Nutrient-flow-based Distributed Clustering (NDC). ROL/NDC uses a combination of routing metrics, which are configured according to the priorities of user-level applications, to improve the network lifetime. Boukerche et al. [27] proposed "QoS Based Routing Protocol" (QBRP), a protocol with service differentiation, in which routes are generated at the actors by using the information collected from the sensor nodes. The next hop for each type of packet is forwarded to the nodes based on the data, which is centrally processed at the actor nodes.

According to the desired characteristics of LRP-QS, presented in chapter 3, a ranking method is used to compare different types of interest traffic in terms of the changes in the observed values. Ranking is an important part of sports. Therefore ranking methods in sports are analyzed to be used in WSANs. Colley [65] and Massey [66] are two of the most important linear algebra-based sports ranking methods with elegant formulations. Masseys model depends on the rule that the difference in the ratings of two teams represents the point differential in a matchup of these two teams. Different than Colley method, Massey method utilizes actual game scores and homefield advantage. There are also examples of methods used in sports ranking, which use Markov methods ([67], [68]). In a recent study, Chartier et al. [69] analyzed Colley, Massey and Markov methods in terms of sensitivity. The results of the analysis show that Colley and Massey methods are less sensitive than the Markov ranking methods to small changes.

2.2 Localization

Localization in WSNs and WSANs has been attracting significant interest in the last decade due to the realization of low-cost and multifunctional sensor nodes and their deployments in both indoor and outdoor environments [70]. Although the localization is significantly important in most of the deployed systems, implementation of the algorithms is challenging since the localization techniques in WSANs have their particular constraints. For instance, the deployment areas are generally complex and they have issues such as accessibility, line-of sight, and so on. The measurement method for the positioning algorithm also needs to be selected according to the environment, the limited capabilities of the sensor nodes and the conditions of the scenario. The heterogeneous node structure of WSANs is yet another challenge and requires a clear differentiation in the roles of the nodes in the network.

The localization methods in the literature generally use *a priori* information about the network or the environment such as the positions of some specific nodes [71] or particular locations in the environment where the nodes can be located at [10]. Localization algorithms also make use of diverse scientific approaches such as graph theory [72, 73], multidimensional scaling [74], sequencing [75], distance vectors (DV) [76], computational geometry [77], particle and Kalman filters [10, 78, 79], Gauss-Markov parameter estimation [80], recursive systems [81], sub-area localization [82], range-free schemes [83], viable kernel-based algorithms [84] and distributed Bayesian algorithms [85].

The measurements of different types of sensors on the nodes can also be used by localization algorithms. Sensor nodes can gather various kinds of information when equipped with appropriate measurement technologies. The main types of sensor measurements used in localization methods are time of arrival (ToA) [86], time difference of arrival (TDoA) [87], angle of arrival (AoA) [88], received signal strength (RSS) [89, 90], and distance related measurements [91]. The measurement technique is normally selected according to the requirements of the environment and the scenario. Multiple techniques can also be utilized in a single approach. Wang et al. [92] propose using multiple techniques for an unsupervised indoor localization scheme, which identifies certain locations in an indoor environment with signatures on multiple sensing dimensions. Michaelides and Panayiotou [93] use RSS at the sensor nodes to report binary observations. The events are reported only when the measured signal strengths are above a threshold. These binary observations are used in an estimation algorithm to construct the likelihood matrix.

We adopt a multi-hop positioning approach for our localization algorithm. There are well known multi-hop algorithms such as the distributed hop-by-hop positioning algorithm, APS, proposed by Niculescu and Nath [76]. APS works as an extension of distance vector routing and GPS positioning to provide the approximate locations for all nodes in a network where only a limited fraction of nodes have self location capabilities. An alternative approach by Savvides et al. [94] uses the sensor nodes, after their positions are estimated, as the anchor nodes in a multilateration algorithm. This process is called iterative multilateration, which is also employed by Savarese et al. [95]. The critical observation in these algorithms, which is also true for our algorithm, is that at least three actor nodes are needed to determine the position of a sensor node. Nagpal et al. [96] organizes a global coordination system in the network by estimating the Euclidian distance of a hop. The estimation algorithm uses the number of communication hops of the sensor nodes and the position error is minimized with imperfect distance estimates. The important theoretical results given by Nagpal et al. are the critical minimum average neighborhood size for sufficient accuracy and the limit on the resolution of a coordinate system determined by local communication.

There are also studies on computationally efficient sequential algorithms for localization. In the sequential positioning algorithm for exact distance measurements by Anderson et al. [72], the sensor nodes are processed in a predetermined order. Fang et al. [97] extended this work in their algorithm "Sweeps" and showed that processing sensor nodes in a specific order can be used to characterize the graph theoretical properties of the network. The Sweeps algorithm relies on the inter-sensor distances and works only with exact measurements. Location estimation of sensor nodes is very critical especially when there is no GPS positioning capability at sensor nodes. The Received Signal Strength Routing (RSSR) algorithm by Boukerche et al. [98] is a GPSless scheme, which makes use of greedy forwarding. After receiving a query packet from the sink, a sensor node uses RSSI to forward the packet to the neighbor that received the query with greater signal strength, essentially the neighbor closest to the sink. Oliveira et al. [81] uses two reference points and the known direction of the recursion for position estimation in their GPSless localization protocol. The protocol works with low-density networks and also indicates the node position error based on the distance to the recursion origin.

High mobility of sensor nodes with the currents in a river is a distinct constraint of our localization scenario and it is considered as a challenge in most of the existing localization methods. The realistic modeling of the river and therefore the mobility of sensor nodes is an important part of our approach. There have been studies on the formulation of the current mobility models. For subsurface ocean currents, Caruso et al. [99] proposed Meandering Current Mobility (MCM) model, extended later by addition of random surface motion by Erol et al. [100]. As an important difference compared to the ocean characteristics, the subsurface currents have faster speeds. On the other hand, the force of the winds and the shape of the terrain (rocks, depth/width variations) generate vortexes in a river, similar to oceans where the particles meander similar to the meanders in the MCM model.

2.3 Animal monitoring with WSANs

There are various studies on the deployment of sensor nodes for wild life animal monitoring and animal tracking. Environmental scientists and zoologists have been increasingly using these technologies to collect data from wild terrestrial areas and transmit them to the remote databases [101]. In some of these applications, the sensor nodes are attached to the animals, forming an ad hoc wireless network of mobile nodes [102].

ZebraNet [103] is one of the initial examples of such studies, in which the animals carry custom tracking collars with GPS capability and form a mobile sensor network across a large area. The collars include sensor nodes with global positioning system (GPS) capability, and they form a WSN to monitor the area and record the data. Wark et al. [102] apply sensor network technology to farming and their approach also includes collars worn by animals. This system utilizes both static and mobile nodes measuring the state of a complex, dynamic system comprising climate, soil, pasture, and animals. Naumowicz et al. [104] deployed a WSN on Skomer Island, Wales to improve the investigation of the behavior and spatial ecology of the Manx Shearwater seabirds. The system informs the scientists with high resolution data about the arrival and departure of the birds and the environmental parameters such as temperature or humidity. CraneTracker by Anthony et al. [105] is a sensor platform for monitoring migratory birds and it is composed of a set of sensors, a multi-modal radio, and power control circuitry. The system uses cellular networks during migration and short range, ad-hoc networks in breeding and nesting grounds. Handcock et al. [106] uses a large cattle enterprise to demonstrate the potential for combining GPS collars and satellite images in a WSN. The implemented WSN is used to monitor behavioral preferences and social behavior of cattle in Northern Australia. The sensor nodes are also used to monitor the functioning of the animal body or a particular organ [107]. In this type of applications, the sensors are implanted within the animal body to collect and transmit information by forming a WSN.

Approaches currently used by biologists in ape and monkey monitoring employ wildlife tracking collars [108], camera traps [109] and subcutaneous implants [110] for animal tagging and data collection [111]. Collars and implants provide more granular data compared to camera traps and tracking collars are the most widely used data collectors [111]. The properties of collars vary according to the hardware they are equipped with, such as radio transmitters, GPS receivers and activity sensors. VHF transmitters on these collars require a short distance to transmit the collected data. Hence, personnel and time requirement is high with the current technology used in the field to collect data. Additionally, collars and implants require tranquilization of animals when attaching a data collector and when gathering data from that data collector about the movement of a particular animal [108]. Therefore utilization of a WSAN improves data collection both in terms of time and also effort despite the difficult constraints of natural environment. Another advantage of using sensor networks is the reduced cost compared to other systems, particularly when sensor nodes are equipped with solar technology [111].

Gathering real field data is challenging for studies on animal behaviors. For instance, Fossey [112] conducted the initial studies on the home ranges of the mountain gorillas and their social interactions. Fossey's observations include hand drawn maps of mountain gorilla ranging paths that form the basis for movement patterns of the gorilla troops. The gorilla behaviors and social interactions in troops are analyzed in a few studies [113], [114]. Due to the absence of public domain real data in this area, models for the generation of movement patterns must be developed and used.

A variety of mobility models have been proposed for simulations of animal groups. The Reference Point Group Mobility (RPGM) [115] by Hong et al. describes mobility coherence in the movement of a mobile host, i.e. hosts at different positions head towards the same target. In RPGM each group has an own logical center and similar to the concept of center of mass, the center's motion defines the entire group's motion behavior. The node deployment in RPGM is usually uniformly at random, but any node deployment can be used to approach the reference point. Nodes in RPGM have their own random movement in addition to the group motion. In the Virtual Track model (VT model) [116] by Zhou et al., nodes follow so called *"switch stations"* that are deployed in the map, creating virtual tracks. Group nodes are distributed along the virtual tracks and the individual nodes are deployed in the whole area. The switch stations have features allowing the nodes to split into several groups after leaving the switch station. These aspects can be often found in the mobility of animal groups such as birds or gorilla troops, which split when a new leader founds a new troop. Musolesi et al. [117] proposes a model, which groups collection of nodes together based on social relationships among the individuals. The groups are mapped to a topographical space, including the strength of social ties. A node belonging to a group moves inside the corresponding group area towards a goal using the Random Waypoint model. Groups also move towards randomly chosen goals with random speeds. As in the model of Hong et al. [115], Musolesi et al. [117] also permits changes in the group affiliation based on a particular parameter (sociability factor).

The deployment and mobility models introduced in this work use the "preferential attachment" concept, which is implemented by Borrel et al. [118] for designing the mobility model called Pragma. The preferential attachment was introduced by Barabási and Albert [119] to explain a common property of many large networks, according to which the vertex connectivity follow a scale-free power-law distribution. Pragma assumes preferential attachment to centers of interest, considering that *"individuals"* move towards *"attractors"*, which appear and disappear. Thus, the model describes independent nodes that exhibit a collective behavior. The authors show that Pragma achieves a scale-free spatial distribution in population growth.

Primate groups are considered as the monitored animal societies in our animal monitoring application. Traditional primate relationship analyses focus on dyadic associations. However, all of the members of primates and their interactions as a hierarchical group must be taken into consideration for an efficient understanding of social structure in a primate society [120]. Some traditional studies provide important information on the roles of individuals in primate groups [121, 122]. Results of these traditional studies started to be integrated with modern social network analysis methods and improved by the utilization of extensive data analyses and quantification [123]. Recent studies show that the social network analysis is useful for the interpretation of primate social structure and organization [124, 125].

Kasper and Voelkl [124] emphasize the importance of the quality of relationships in primate social systems and discuss a set of network measures for social network organization. The analysis includes results of 70 primate groups from 30 different species. Clark [125] studies the spatial association and social interaction data collected from a group of zoo-housed primates. Results showed that social network analysis reveals important characteristics of primate groups when proximity among individuals is not forced. Sueur and Petit [126] use movement patterns along with network metrics such as centrality and clustering coefficient to understand the roles, rankings and associations in the social group. Matsuda et al. [127] use similar network parameters as Sueur and Petit [126] to compare the intra-group relationships in primates. Their results show the important potential of contribution that social network analysis has for primate social bond analysis. Flack et al. [128] show the importance of individuals with high clustering coefficient on group stability and conflict management. They use experiment results to quantify instability of group structure in terms of reduced mean degree, increased clustering, reduced reach, and increased assortativity.

2.4 Node positioning in aerial networks

The literature on 3D wireless sensor networking mostly focuses on coverage problems. In the conventional 2D scenarios, a sensor coverage is generally modeled as a circle and the maximal coverage problem is mapped to a circle packing formulation which has a polynomial time solution. This problem turns into the sphere packing problem in three dimensions. Given the high complexity of the sphere packing problem for 3D coverage, Alam and Haas [129] argue that space filling polyhedrons would be more suitable and try to fill the 3D application space with the least number of polyhedrons in order to provide maximal coverage. Ravelomanana [130] studies the properties of the network topologies that result from random deployment of nodes in a 3D region of interest to provide theoretical bounds that can help in preliminary design and feasibility studies of 3D WSNs. The author derives conditions for the node transmission range r required for achieving a degree of connectivity d, where every node has at least d neighbors. Pompili et al. [131] uses Ravelomanana's bounds to validate the effectiveness of their 3D random underwater node deployment scheme. Zhou et al. [132] present two algorithms for discovering boundary nodes and constructing boundary surfaces in 3D wireless networks. Bai et al. [133] designed and proved the optimality of one and two connectivity patterns under any value of the ratio of communication range over sensing range, among regular lattice deployment patterns. The authors also introduced three and four connectivity patterns and investigated the evolutions among all of the proposed low-connectivity patterns. Slab Routing by Chiang and Peng [134] adapts 2D geographic face routing techniques to 3D space by dynamically creating a space partition and executing face routing over the planar projected graph of nodes contained within.

There are aerial sensor network implementations for various applications concerning the measure of air pollution and weather variables. Only just a few experiments for volcanic gas sampling exists in the literature, while the autonomous aerial system by Astuti et al. [135] has a very similar application scenario with APAWSAN. In their system, there is no network but a single UAV, which performs aerial surveillance of volcanic areas and to analyze the composition of gases inside volcanic plumes. The SensorFly system [136] by Purohit and Zhang is a mobile-controlled flying sensor network that monitors changes in a dangerous environment such as an earthquake or fire. SensorFly uses a flying miniature sensor with a weight of 30g and low mass production cost around $100. Elston and Frew [137] [138] presents a hierarchical control architecture similar to APAWSAN with a mother-ship, which acts as a distributed database and daughter-ship micro air vehicles, which use vector field tracking. Autonomous Flying Robot MARVIN (Multipurpose aerial robot vehicles with intelligent navigation) project [139] uses robots with the ability to coordinate with each other to complete required tasks. SensorFlock by Allred et al. [140] is an airborne WSN composed of bird-sized micro aerial vehicles (MAVs), with a focus on the design of the MAVs and received signal strength indication (RSSI). Their WSN is composed of hundreds of inexpensive, semiautonomous, and cooperating airborne vehicles making observations and relaying data over a wireless communication mesh network.

We apply our positioning strategy to real-world scenarios by utilizing a rotatable hybrid antenna model. The communication reliability advantages of directional antenna have been discussed in literature (see [141]). Jiang et al. [142] demonstrate a localization scheme using beacon nodes with directional antennas, which rotate regularly. After evaluating the received signal strength indication (RSSI) values of the beacon signals, a sensor node estimates the orientation relative to the beacon node. However, this approach works for the 2D static or mobile sensor networks but not for 3D space.

Beam steering based [143] and scan-based [144] algorithms with directional antenna are also proposed. Beam steering works for infrastructure based networks. It is not practical for UAV networks while scan-based approach requires all nodes to follow the same search sequence. Adaptive Medium Access Control protocol for UAV (AMAC_UAV) [145] is designed for a network of UAVs with directional antennas. Each UAV is equipped with two directional antennas for data transmission and two omnidirectional antennas for location packets. Data transmission via omni antenna is limited by the transmission range of the antenna. As the number of antennas increases, the complexity of the algorithm and the power requirements also increase. Moreover, the data transmitted by omni antenna has a high probability of packet loss, especially when the node is mobile.

The sensor nodes and actors are positioned on UAVs in our application scenario for 3D positioning. UAVs are used for various purposes including military ([146, 147]) and science [148] applications as well as disaster monitoring ([149–151]) while the industry is growing fast and doubled in the last decade according to market studies. Furthermore, there are companies and corporations providing custom-made vehicles with various capabilities. In the survey of autopilot systems for small UAVs by Chao et al. [152], small-sized UAVs are defined to be light-weight with shorter wingspans. However small-sized UAVs are also cheap and expendable since their development and operation are easier compared to large UAVs. Dempsey [153] grouped unmanned aircraft systems (UAS) into five categories according to their capabilities, advantages, and limitations. Group one UAS are typically hand-launched small vehicles capable of altitudes less than 1200 feet above ground level, while Group five UAS are the largest systems with extended capabilities in terms of endurance, air speed, range, and altitude.

For 3D positioning, we make use of the VSEPR theory of chemistry, which is based on the idea of a correlation between molecular geometry and the number of valence electrons around a central atom. This concept was first presented by Sidgwick and Powell [154]. Gillespie and Nyholm [155] refined it later and built the elaborate VSEPR theory, which states that the maximum repulsion of the electron pairs or atoms defines the geometric optimum positions of peripheral atoms or alone electron pairs that maximizes the distance between these entities.

CHAPTER 3
LIGTHWEIGHT ROUTING WITH DYNAMIC INTERESTS

In this chapter, a lightweight routing protocol with QoS support (LRP-QS) [12], [13] is introduced. The goal of the protocol is to provide the highest rate available for each flow and to dynamically prioritize the interests according to the observed values in a WSAN. WSAN considered in the application scenario consists of stationary sensor and actor nodes distributed in an area. Actor nodes have no initial information on the positions of the sensor nodes. Since the number of actors are low compared to the number of sensor nodes, the sensor nodes, which are not directly connected to an actor, communicate with other nodes according to the requirements of the network. Hence, the network is organized for an efficient data transmission.

In the context of this work, QoS is defined as the assurance of services required by the applications. The main optimization objectives are the packet transmission and loss rates. At sensor nodes, our approach is lightweight and efficient in terms of memory and power consumption as opposed to actors which have a more complicated task in accordance with their capabilities. The sink expresses a set of interests regarding the information to be collected. The interests are transmitted to the actors through the communication backbone with initial weights, which define their rate requirements. According to the events observed, these interests are ranked at each actor to adjust their weights according to the network conditions. The application scenario also includes bursty data sources resulting in congestion at various points in the network from time to time, which increases the criticality of efficient resource utilization in the network.

LRP-QS is motivated by QBRP [27] and shares several aspects with QBRP in network organization. We identify disadvantages of QBRP and aim to improve the routing performance in LRP-QS. QBRP has a routing algorithm with a high computation and communication cost. The data collection from the sensor nodes and updating them with path and interest information require excessive energy and communication. Additionally, if there is no packet transmission on a node, that node or the path cannot be updated. One of the main criteria for routing in WSANs is energy consumption. QBRP aims to use less energy expensive path. On the other hand, high memory usage and frequent updates result in high energy expenditure, which makes it very costly for implementation in WSANs. The LRP-QS overcomes these problems as described in the following sections.

3.1 Network organization

3.1.1 Actor areas

The network configuration starts with the flooding of area configuration packets (ACP) by the actor nodes. An ACP includes the actor's ID and the number of hops the packet is forwarded. Sensor nodes have *actor ID* and *hop value* attributes and neither of them has initial values. In other words, initially a sensor node knows neither the actor node it is associated with nor the hop-distance to that actor node.

When a sensor node receives an ACP, it first checks the *hop value*. If the value is greater than or equal to the node's hop value, the packet is dropped. Otherwise the node updates its attributes with the values in the related fields of the packet and retransmits the packet. Thus, the sensor node keeps the information for only one actor node even when it receives ACPs from multiple actor nodes. The node keeps the address of the neighbor node it received its actor's ID as the destination for its data packets. The node can keep multiple destinations and this record can be leveraged to have more control on resources. However one of the main objectives is to keep the protocol as lightweight as possible on sensor nodes. In addition, the hop value of a sensor node automatically changes as one of its neighbors fails since all communication is handled locally.

When the actor areas are formed, each sensor node is affiliated with an actor, has information about its lower-hop neighbor(s) and the number of hops required to reach its actor. Figure 3.1 shows the hypothetical view of the network with the connections after this phase, where each actor area is presented with sensor-sensor and sensor-actor connections.

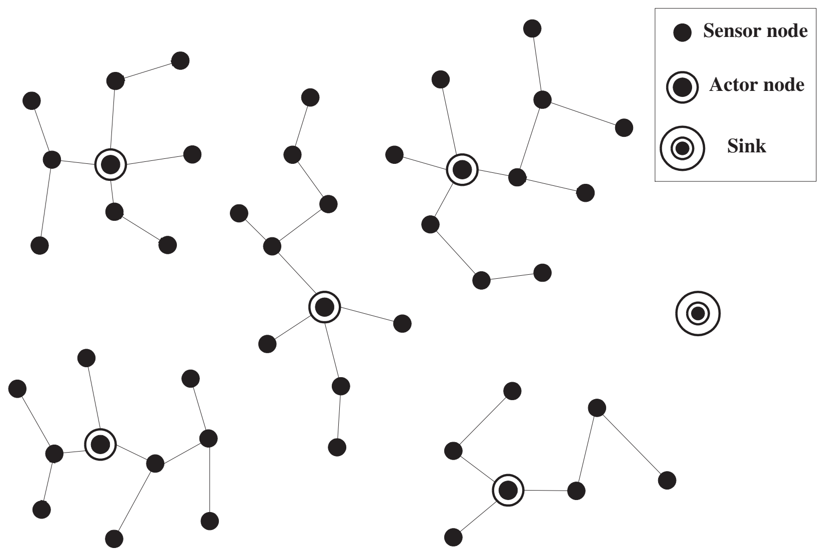


Figure 3.1: The network view after formation of actor areas.

3.1.2 Communication backbone

The actor areas must be connected to transmit the collected data to the sink. We define the network formed by the links among actors and the sink as the "communication backbone". The sink starts formation of the *communication backbone* by sending an area integration packet (AIP) with its ID in the source field.

The destination address for data packets (DA_d) is an attribute of each actor. When an actor receives an AIP from the sink, it saves the sink as the destination address for data packets (DA_d). Then the AIP is forwarded in the network among actors. Hence the sink is positioned in the transmission range of at least one actor in order to prevent bottlenecks at the links close to the sink. Otherwise the sink would receive the collected data through sensor nodes, which would create severe packet loss and delay.

If an actor doesn't receive an AIP from a sink, the first actor from which it received the AIP is recorded as the DA_d. Then the actor places its ID on the AIP and retransmits it. If an actor receives AIPs from multiple actors, it saves the extra actor IDs in the "redundancy list" (L_r). This list is kept at an actor for future use in case of a change in the communication backbone such as a dead actor node. A summary of the pseudocode of the algorithm used at each actor node receiving an AIP is presented in Algorithm 1.

If each actor is guaranteed to be in the transmission range of at least one actor, i.e. the network communication backbone is connected, then the AIPs do not need to be processed at sensor nodes. Figure 3.2 shows the network view with actor areas and a connected communication backbone.

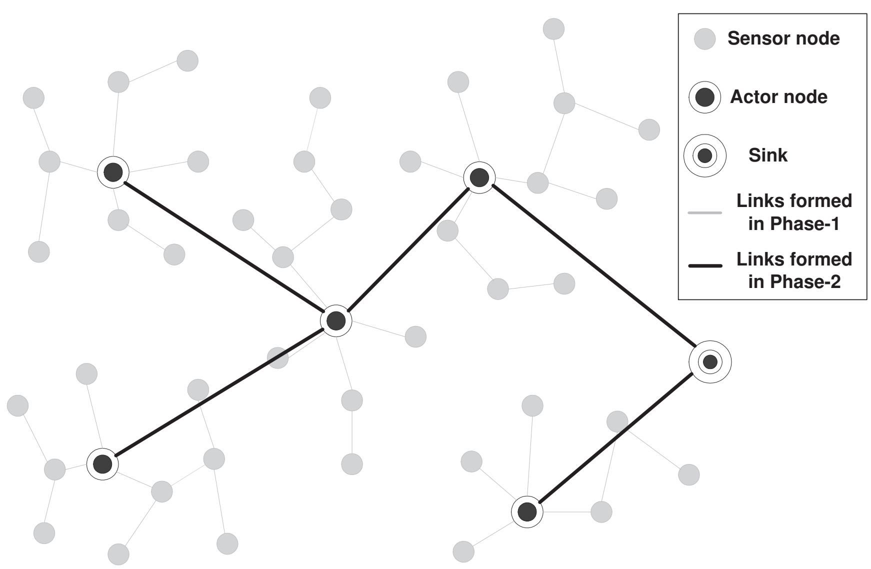


Figure 3.2: Network view after actor areas and network among actors are formed.

Algorithm 1 Processing of AIP at an actor node

Check source address of AIP
Check DA_d
if DA_d is a sink **then**
    drop AIP
else if AIP is received from a sink **then**
    DA_d = ID of the sink transmitted AIP
    transmit AIP
else if AIP is received from an actor **then**
    if DA_d is an actor **then**
        Put ID of the actor that transmitted AIP in L_r
    else
        set DA_d = ID of the actor transmitted AIP
        transmit AIP
    end if
else if AIP is received from a sensor node **then**
    if DA_d is an actor **then**
        drop AIP
    else
        DA_d = ID of the actor on AIP
        transmit AIP
    end if
end if

In the case when there is no actor nodes in the transmission range of an actor node, sensor nodes, which receive AIP of this actor node, use a lightweight algorithm to process these packets. The receiving sensor node checks if the actor ID on AIP is equal to the node's associated actor node ID. The packets with matching actor node ID are transmitted to the sensor node's higher-hop neighbor(s) and the first AIP from another actor is transmitted to the lower-hop neighbors in order to guarantee the conveying of AIPs in the network. All other AIPs are dropped by the sensor nodes.

3.2 Data collection

3.2.1 Interest subscription

The interests are distributed to the actors via the communication backbone. When an actor receives an interest from the sink, it checks and updates its interest subscription table with the received interest unless the interest is already included in the table. All the information expressed in the interest packet such as type or time is stored in this table. Since actor nodes have larger memory resources and processing capabilities compared to sensor nodes, they keep the information received from the sink about an interest for certain periods.

Each actor transmits the interests to the closest sensor nodes to start the selective flooding (i.e. forwarding only the packets from higher hop neighbors) of the interests in its area. When a sensor node receives a new interest, it updates its subscription table. If the sensor node observes an event that the sink is interested, then it will generate data packets. However if a node does not sense an event but it is only on the path of data transmission, then it will keep the interest in its subscription table and use this information when routing the data packets.

At the end of this phase, each sensor node knows which type of packets to generate and which events to generate packets for. We define the interests, which the sensor nodes generate packets for, as the *active interests*. Each sensor node also has the interests in their subscription table for which they reside on the path from the reporting sensor node to the actor. This type of interests is defined as *passive interests* and sensor nodes keep only on-off information for these interests.

3.2.2 Ranking interests

The requirements of the network are distributed by using the interests. However this approach of Intanagonwiwat et al. [7] have disadvantages for a network deployed to collect information from an environment for a long time. In this approach, the interests of the sink can be changed only by sending new interests. Even if the collected values stay constant for an event type with an initially high weight, that event type stays still important unless the collected values are actively observed and the interests are changed.

We consider a more dynamic behavior to create a self-organizing protocol, in which the change in the observed values for an interest defines the importance of the interest. Our system changes the interests dynamically according to the events occurred. Therefore as the events are observed, our system also ranks the interests to increase the quality of observation and to reduce the traffic and energy expenditure.

The ranking method of our approach is based on the amount of changes in the observed values for events of interest. Consequently the importance of the interests is determined according to the percentage changes in their values. For instance if the packets received in response to interest A in a predefined period of time have high fluctuations in their values compared to the constant values of interest B packets, then A is considered as more important than B for that *time period*. It is important to note that rankings of interests will differ at each actor area as the interests are ranked at each actor depending on the events locally in that area.

3.2.2.1 Ranking requirements

An interest, which had no change in its observed values for a long time, must not gain high importance suddenly when there are events with fluctuating values for that interest in only one *time period*. In other words, a sudden increase in the value of a packet in response to an interest must not be very effective on the ranking of the interests. In other words, the ranking method must not be sensitive. Therefore the sensitivity of the method is critical when choosing the method for ranking interests, which eliminates some of the most popular ranking algorithms.

Chartier et al. [69] analyzed the sensitivity of Colley, Massey and Markov methods to small perturbations and determined how much the ranking is affected by these changes. Similar discrepancies in the input and output ranking data showed instability of the ranking methods. The authors concluded that while the Colley and Massey methods are insensitive to small changes, the Page Rank method is highly sensitive to such changes, often resulting in anomalies in rankings. While it is a desirable property in web page ranking, it does not apply to sports or our approach. A single loose should not change the whole ranking list or result in a rank jump. Colley and Massey methods have an isolated response, resulting in changes to the rankings of only two objects in question. Colley method is based only on results from the field, which is more appropriate for interests since metrics such as homefield advantage have no correspondence in our approach. Therefore Colley method is chosen over the Massey method for our application.

3.2.2.2 Ranking by Colley based method

The main goal of Colley Matrix Method [65] is producing "fair" rankings in sports. It has been used as a part of Bowl Championship Series (BCS). Wins and losses of the teams are the only input information used in this model. Mathematical fundamentals of Colley method are described in this section. When describing the method, the parameters that our approach used and their equivalent terms in sports are given.

When the percentage change on the value of an interest is more than the change in another interest, it will be considered as a win of the former over the later one and the number of wins of an interest i is defined as $n_{w,i}$. $n_{total,i}$ is the total number of comparisons made, which corresponds to the number of games in sports. In traditional ranking methods, the winning ratio, which is defined as $\frac{n_{w,i}}{n_{total,i}}$, is used. Colley method instead defines a modified rating of r_i for interests, which is defined as follows [156]:

$$r_i = \frac{1 + n_{w,i}}{2 + n_{total,i}} \tag{3.1}$$

The advantage of using this equation for the definition is to avoid the inconsistency in comparison of interests in specific conditions. For instance, the interests with no changes or the interests with changes in every *time period* would have values of 0 and 1 respectively if traditional win ratio is used instead of defined r_i. In our application, the interests can have initial values or the weights of the interests can be initially zero and dynamically modifiable. When interests have no initial weights after the first *time period*, the interest with a change is "infinitely better" than the interest without a change. Using the Colley method, the former interest ($r = \frac{2}{3}$) would have a twice the better score compared to the later interest ($r = \frac{1}{3}$). Hence, the initial rating of any interest with no changes is equal to $\frac{1}{2}$, which is the median value between 0 and 1. Depending on the comparisons, a win increases and loss decreases the value of r. This approach results in a system less sensitive to changes. In order to adjust the performance measure to the weight of the other interests, we transform the values of n_w as follows:

$$n_w = \frac{(n_w - n_l)}{2} + \frac{n_{total}}{2} = \frac{(n_w - n_l)}{2} + \sum_{j=1}^{n_{total}} \frac{1}{2} \tag{3.2}$$

Instead of considering the actual number of wins, the effective number of wins $n_{w,i}^{eff}$ w is calculated by adjusting the second term of the expression, which represents the summation of n_{total} terms equal to $\frac{1}{2}$ corresponding to the default rating of an interest without a comparison. For taking the strength of the other interests into account, these terms are substituted by actual ratings of the interests r_j and the formula for the effective number of wins for an interest i is as follows:

$$n_{w,i}^{eff} = \frac{(n_{w,i} - n_{l,i})}{2} + \sum_{k=1}^{n_{total,i}} X_{ijk} r_j \tag{3.3}$$

Combining these formulas, the linear equation relating the ratings of an interest and the others is written as:

$$(2 + n_{total,i}) r_i - \sum_{j=1}^{n_{total,i}} X_{ijk} r_j = \frac{(n_{w,i} - n_{l,i})}{2} \tag{3.4}$$

If the total number of interests at an actor is equal to N, then the equations of this form will be written for all of them, which results in the linear system with N equations and N variables:

$$C\vec{r} = \vec{b} \tag{3.5}$$

where $\vec{r}$ and $\vec{b}$ and C (Colley matrix) are defined as follows:

$$\vec{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 1 + (n_{w,1} - n_{l,1})/2 \\ 1 + (n_{w,2} - n_{l,2})/2 \\ \vdots \\ 1 + (n_{w,N} - n_{l,N})/2 \end{bmatrix}$$

$$C = [c_{ij}]_{i,j} = 1 \ldots n \tag{3.6}$$

The elements of Colley matrix are defined as follows:

$$c_{ii} = 2 + n_{total,i} \tag{3.7}$$

$$c_{ij} = -n_{j,i} \tag{3.8}$$

where $n_{j,i}$ is the number of times the interests i and j compared to each other.

The matrix C is positive definite [65], which allows the efficient solving of the linear system of C, $\vec{r}$ and $\vec{b}$ using standard techniques. Cholesky decomposition can be used with back-substitution to solve this linear system of equations. The solution of this system would represent the vector of numbers corresponding to the ratings of all N interests, and the resulting rankings are determined by sorting the elements of the solution vector r in decreasing order.

When adapting the Colley method for ranking interests, there are two important constraints: the duration of a period and the minimum change in the sensed values to update weights.

Our system continuously collects information from the environment. The duration of the *time period*, which will be used to compare interests, depends on the particular requirements of the systems. Thus, the elements of Colley matrix become:

$$c_{ii} = 2 + \frac{T_t}{t} \qquad\qquad c_{ij} = -\frac{T_{j,i}}{t} \tag{3.9}$$

where T_t is the total time passed, $T_{j,i}$ is the total time that both interest i and interest j are active and t is the chosen *time period*.

The actors update the weights of the interests in its records according to their calculated rankings. However if the weights are updated at the actor areas for every change in the ranking values of the interests, these updates can create excessive network traffic. Therefore a range (l) is defined to update the network as follows:

$$l = \frac{\sum_{i=0}^{N} \left(w_i \left(\frac{T}{t}\right) - w_i \left(\frac{T}{t} - 1\right)\right)}{N} \tag{3.10}$$

where w_i is the weight of the interest i.

There is a trade-off between the value of l and the number of updates for interests. If we choose a very small value for l, there will be high responsiveness to the changes in observed values with the requirement of frequent updates in each actor area. If l is chosen to be large, then the number of updates decreases. However, if l is very large, the responsiveness to changes will be low and the system will essentially behave as if the weights are predefined.

3.2.3 Data transmission

Traditional approaches achieve QoS support in terms of rate guarantees provided to different flows by keeping detailed state information for each flow. The state information includes the expected rate of the flow, the real rate of the flow, update time for the information, and the time window to make decisions. However, this approach is not feasible with low memory and energy resources of sensor nodes. In order to eliminate the per-flow state and high computation requirements, we use a method based on the approach of Stoica et al. [157], which can be described as "having packets carry the state".

The number of bits in the *rate* field of the packet denotes the rate of the packet R_p. There are state encoding mechanisms in the literature by which large numbers can be represented by small number of bits. Therefore each state variable can be restricted to a predefined number of possible values to minimize the complexity of per packet processing. For instance, eight bits ensure 256 (2^8) different values to assign for the rate of a packet at a sensor node.

When an event is captured by a sensor node, the node checks its subscription table. If there is an active interest for that event, the sensor node generates a data packet to notify the actor. The nodes, which capture events in the areas of interests, start the reporting of the events. The pseudocode of the routing algorithm at each sensor node is presented in Algorithm 2.

Algorithm 2 Routing in a sensor node

```
N_f: Number of flows on the node
N_g: Number of greedy flows on the node
C_r: Remaining output capacity of the node
C_s: Output capacity used by the sharing flows
R_e: Fair rate for flows on the node
P_d: Dropping probability of a packet
if a packet is received by a sensor node then
    if the packet is a notification to an interest then
        if it is the first packet for that interest then
            set F_s for the interest
            increment N_f by 1, reduce C_r by R_p
        end if
        if C_r > 0 then
            forward the packet
        else
            if R_p > R_e then
                set the F_g of the interest
                if the F_s of the interest is unset then
                    increase C_s by R_p, set F_s of the interest
                end if
                drop the packet with its P_d
                if the packet is not dropped then
                    fill the rate field and forward the packet
                end if
            else
                forward the packet
                if the F_s of the interest is set then
                    reduce C_s by R_p
                end if
                unset F_g and F_s of the interest
            end if
        end if
    else
        if the packet is not in response to an interest then
            drop the packet (unwanted packet)
        end if
    end if
end if
```

Sensor nodes have a predefined maximum transmission capacity called "output capacity" (C_o). When an intermediate sensor node receives a data packet in response to a passive interest, it switches this interest's state to "on". The total number of interests with state "on", (N_f), represents the number of interests sharing the output capacity of this node. If it is the first received packet for that interest, the output capacity is reduced by the rate of the packet and saved as the remaining output capacity (C_r) of the node.

3.2.3.1 Packet processing

$C_r > 0$: The received packet doesn't require any further processing or encoding unless the output capacity is exceeded.

$C_r < 0$ and $R_p > R_e$: Packet drops occur when C_r becomes negative. We define the efficient rate (R_e) as the amount of output capacity that the node can fairly employ for a flow when C_r is negative. R_e is formulated as follows:

$$R_e = \frac{C_o}{N_f} \tag{3.11}$$

If the rate tag on a data packet is greater than R_e, this means the packets of the interest are received with a rate greater than the rate shared for that interest at that node. Hence this interest will be tagged as "greedy" by setting the greedy flag (F_g) of the interest in the subscription table. The number of packets to drop and the method to drop these packets must be determined in order to provide an efficient service to data traffic. It's important to note that R_e is the maximum value a packet will be encoded with when all flows are received with rates greater than the efficient rate.

There may be packets received with rate values lower than R_e when C_r is negative. In such a case, if all packets are encoded with rate values smaller than or equal to R_e, there will be an excess capacity that is not used. Our algorithm is designed to use this excess capacity since efficient usage of resources is critical in QoS. Furthermore, our main constraint is the rate of data flows and higher rates for flows can be achieved by employing the excess bandwidth. Accordingly, we define the *shared capacity* C_s, the capacity shared among flows that are received with rates greater than R_e. The interests using C_s are defined as *sharing interests*.

The inserted rate value represents the estimate of the flow's incoming traffic at the next node and it also depends on the importance of the packet. The priorities of interests are specified by their weights. These values are expressed by the sink when registering the interests and more important interests are registered with higher weights. Therefore the new R_p depends on the weight of its interest w_p and the number of sharing interests N_s and it is calculated as:

$$R_p = \frac{C_s \cdot w_p}{\sum_{i=0}^{N_s} w_i} \tag{3.12}$$

$C_r < 0$ **and** $R_p < R_e$**:** If R_p of a packet is lower than R_e when C_r is negative, that packet is forwarded without replacing its rate tag. The interest doesn't share C_s with greedy interests, so its flags are unset. An ungreedy interest can become greedy after new packets for different interests are received by the node and since it uses C_s after that instant, its rate value is added to C_s.

In order to insert an exact rate value in the packets, number of transmitted and dropped packets must be recorded at the sensor node for a period of time, which is not efficient with limited resources of the sensor nodes. Therefore packet dropping is done probabilistically at sensor nodes using the rate tags, subscription tables and the output capacities. Dropping probability of a packet increases as the difference between the calculated new rate and the rate tag gets larger. The probability to drop a packet is defined as follows:

$$D_p = 1 - \frac{C_s}{N_s \cdot R_p} \tag{3.13}$$

In contrast to sensors, actors keep state information for each flow. In order to estimate the flow arrival rate, we use the following equation, which depends on the exponential averaging formula in Stoica et al. [157].

$$R_i^{new} = (1 - e^{-T/K})\frac{l}{T} + e^{-T/K} R_i^{old} \tag{3.14}$$

where T is the time between the last two packets of the interest i, l is the packet length and K is a constant. The actor nodes insert these flow rates on each packet they transmit. Using an exponential weight $e^{-T_i^k/K}$ provides more reliable estimation for bursty traffic, even when the packet inter-arrival time has significant variance.

Our application scenario is advantageous for using the approach of Stoica et al. [157] compared to traditional internet routing scenarios. In traditional applications, packets are labeled at the edges of the network by using an estimation algorithm and these labels are updated at the core with a probabilistic approach until the packet reaches a boundary of the network. However in our scenario, a packet is injected into the network with the exact rate placed on its label. Then the tag is updated as the packet is transmitted to the closest actor. An actor has information regarding the flows in its interest subscription table. There is no extra need for exponential averaging to estimate the fair share rate at the actor node, which is calculated using the rate and weight values expressed by the sink. By means of the high transmission range of actor nodes, the data is transmitted to the sink via a path formed by actor nodes.

3.3 Simulation study

3.3.1 Simulation environment

The simulations are carried out in OPNET modeler [158] to analyze the performance of the protocol. All nodes in the system are stationary and in each simulation, a network topology is generated with the a sink, and 60 sensor nodes distributed randomly over the entire area ($200x200m$) with four actors at predefined locations. IEEE 802.11 is used as the underlying MAC layer with direct sequence physical characteristics, $8.02 \cdot 10^{-6}$ Watts transmission power, -95 dBm packet reception power threshold and auto assigned channel settings of OPNET modeler. Transmission ranges of sensor and actor nodes are taken as 50 and 180 meters, respectively. The data packet size is constant and 256 bytes.

The performance of the protocol is evaluated by using the simulation metrics packet loss, control overhead, memory consumption and end-to-end delay in the first set of experiments, in which our protocol is compared to QBRP [27]. Simulation scenarios are chosen similar to the ones used by Boukerche et al. [27] when evaluating the performance of QBRP. We also have additional scenarios for evaluating the performance under certain conditions.

3.3.2 Simulation results

3.3.2.1 Packet loss

In order to create data traffic, eight event sources are placed in the field such that each actor area has two event sources, producing interest 1 and 2 events with equal priorities. Interest 2 packets are injected into the network five times more than interest 1 packets. Since QBRP does not define a method for ranking interests, our protocol also worked with constant interest weight values in these experiments.

Figure 3.3 denotes the packet loss with increasing event generation rates. Significantly less number of packets are dropped compared to QBRP. As the number of dropped packets decreases, the delivery rate and the reliability of the protocol increase. Additionally, Interest

1 packets are protected while QBRP is not able to do so. Since both types of the traffic have equal priorities, our algorithm tends to drop Interest 1 packets much less than greedy Interest 2 packets. QBRP drops Interest 1 packets almost twice as much whereas the ratio is much less when we compare dropped Interest 2 packets.

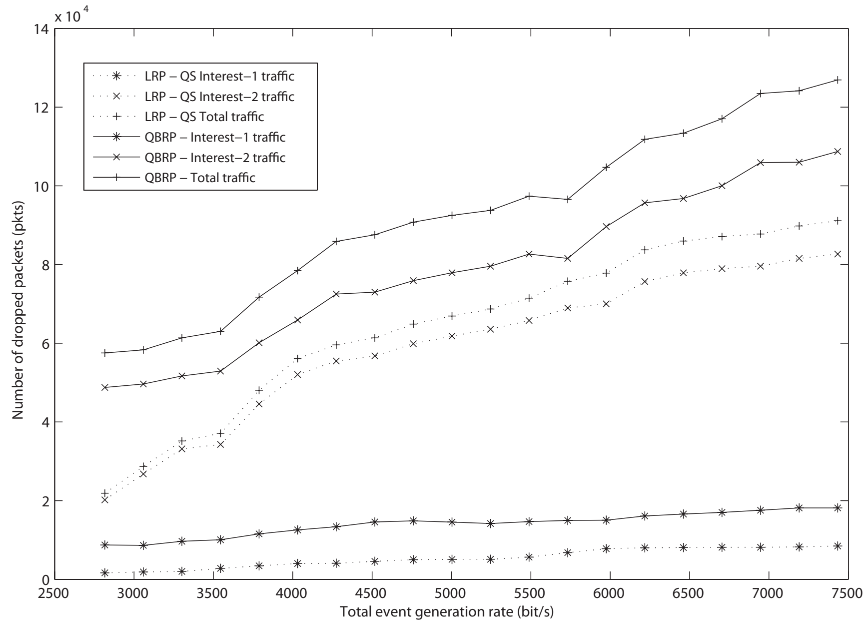


Figure 3.3: Packet loss

3.3.2.2 Control overhead

Number of control packets is critical since increased traffic means more delay and energy consumption. Boukerche et al. [27] showed that number of control packets used by QBRP is not critically affected by the rate of packet generation. Hence we monitor the average number of control packets used by the protocols with non-increasing packet generation rates but varying number of traffic types. Figure 3.4 shows that the proposed protocol outperforms QBRP, by using 45 percent less control packets on average.

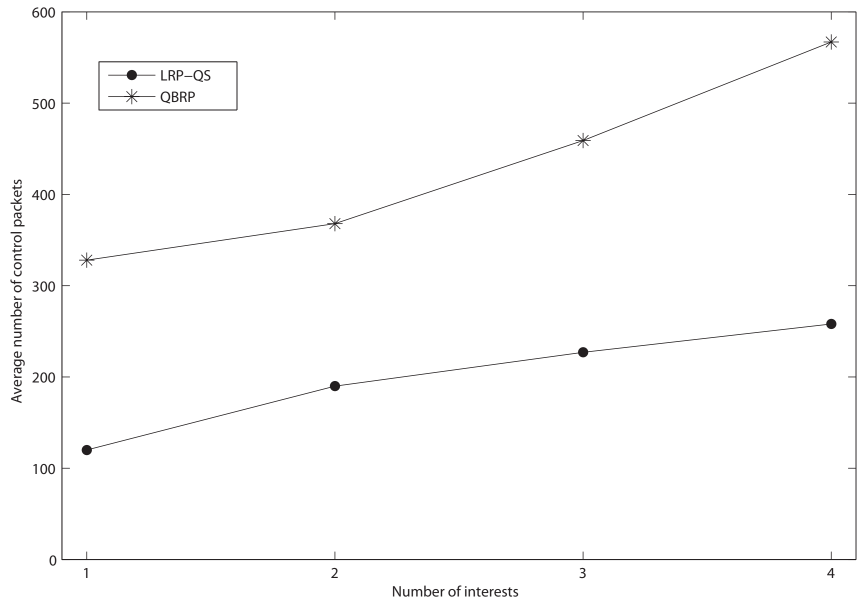


Figure 3.4: Control overhead

3.3.2.3 Memory and energy consumption

The memory consumption is defined in the simulations as the total memory consumed by all nodes. Figure 3.5 denotes memory consumption ratio of the protocols with increasing number of interests. Sensor nodes are deployed randomly in the area in each scenario while the positions of the sink and the actor nodes remain the same throughout the simulations. Our protocol uses less than half of the memory used by QBRP in 95 percent of all cases and also performs better with increasing number of interests. QBRP's memory requirement grows with an increasing rate with each additional interest type.

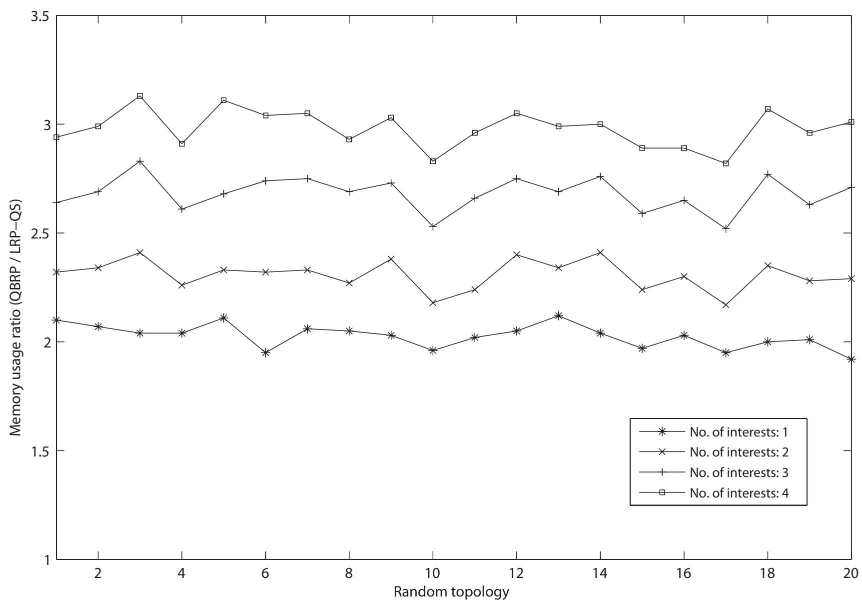


Figure 3.5: Memory consumption

Figure 3.6 shows the mean of the ratio and the range for the ratio values observed in the simulations for each number of interests. We find that our protocol uses less than half of the memory used by QBRP in most cases and performs better with increasing number of interests. As the number of interests exceeds two, the range between minimum and maximum values for memory usage ratio increases.

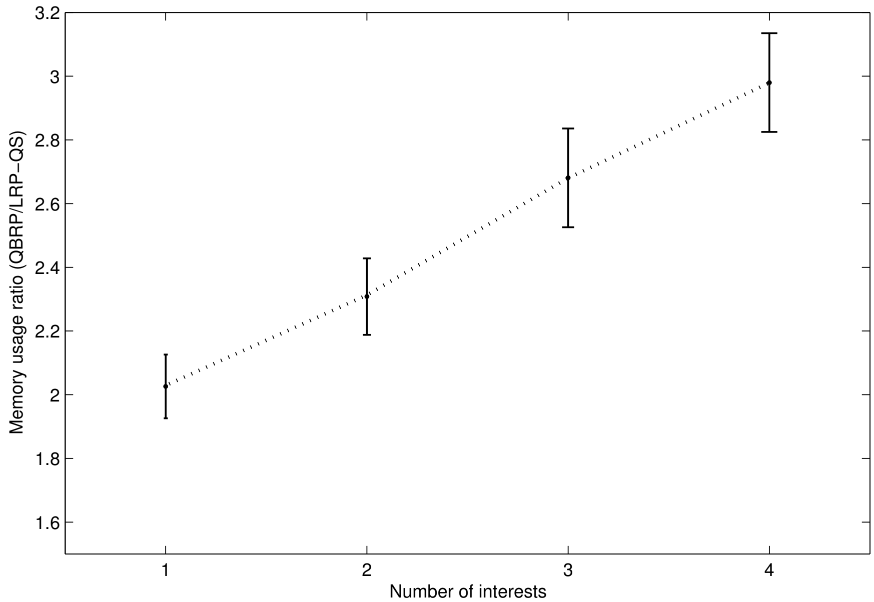


Figure 3.6: Memory consumption ratio

The memory consumption of QBRP depends highly on the locations of interests and the sensor density in these locations. The probability of having different values for these metrics increases with increasing number of interests. Therefore, the ranges for the ratios slightly increase as the number of interests exceeds two. In correlation with the results on control packets, the energy consumption of QBRP is greater than our protocol for all cases and the ratio increases with the increasing number of interests as shown in Figure 3.7.

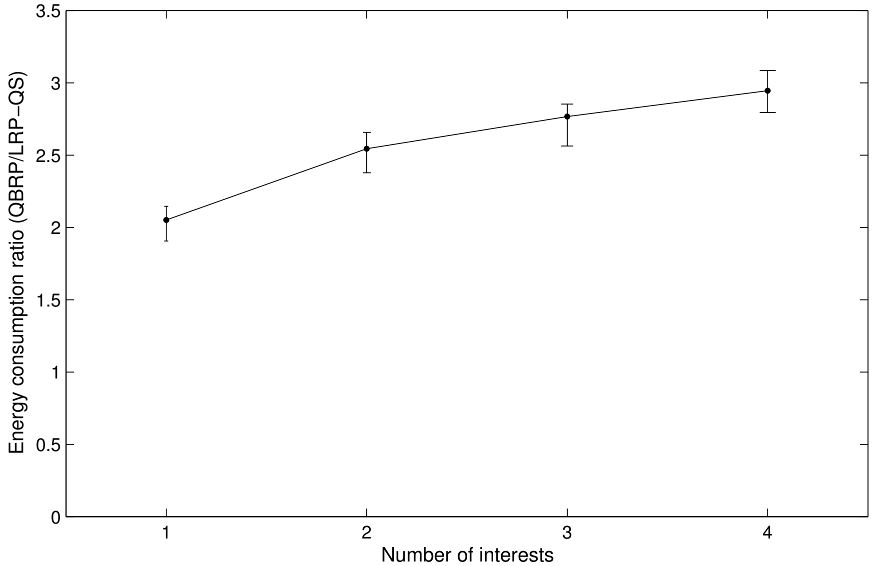


Figure 3.7: Energy consumption ratio

3.3.2.4 End-to-end delay

Figure 3.8 shows end-to-end delay values with increasing number of events for our protocol and QBRP for each number of events with 95% confidence interval. QBRP selects paths efficiently with extensive data processing and memory consumption. However central processing and the route configuration messages cause QBRP to generate extra traffic as we have seen in previous simulation results. Our protocol achieves a delay performance similar to QBRP without the increased control overhead, memory consumption or packet drop rate.

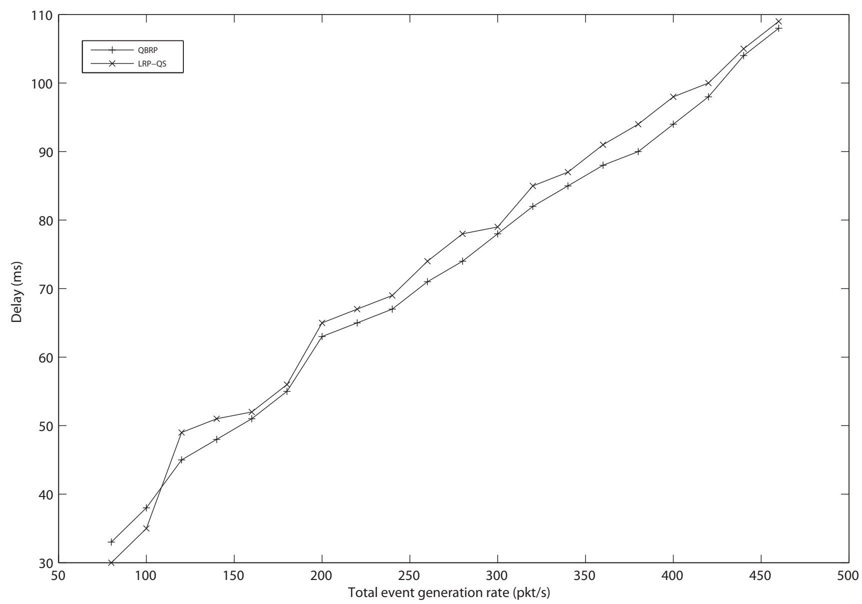


Figure 3.8: End-to-end delay

Figure 3.9 presents the average end-to-end delay values with increasing number of sensor nodes for LRP-QS and QBRP with a constant total event generation rate of 200 pkt/s. The protocols have similar delay results for different network sizes and the delay performances of both protocols improve as the number of sensor nodes increase. This result is due to the new communication hops formed by the additional sensor nodes. LRP-QS dynamically adapts to changes and performs better as the new paths are formed. The main sources of delay in QBRP are central processing and route configuration messages, which create extra traffic even when the network size is small. Therefore, the percentage change in the average delay of our protocol is slightly higher than QBRP as the number of sensor nodes increase.

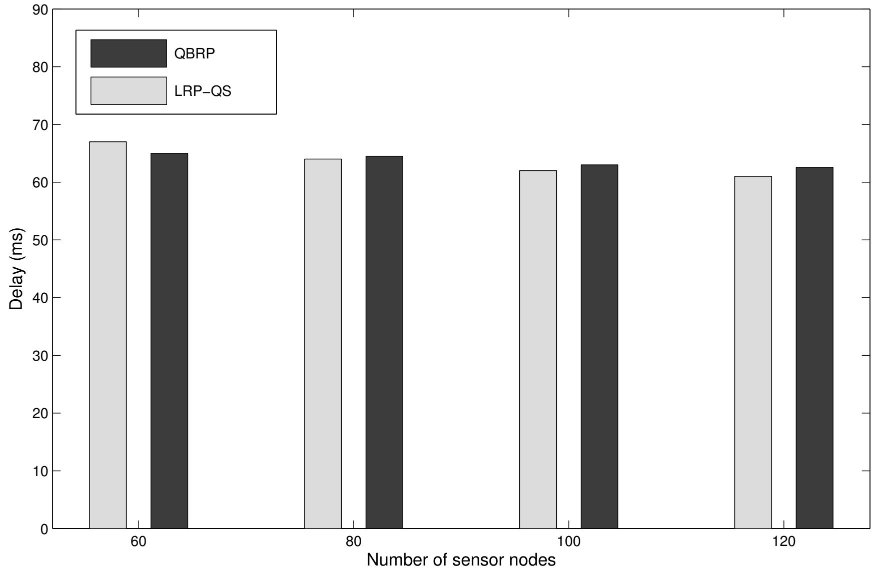


Figure 3.9: End-to-end delay for different network sizes

3.3.2.5 Ranking

The main aim of ranking is assigning weights according to the changes in the values of the collected data at the actor in response to the interests.

In the first set of experiments, performance of the ranking algorithm is evaluated in an actor area. There are four types of interests, which have equal initial weights. The event sources are placed in the area such that interest 1, interest 2 and interest 3 packets are produced continuously with values in a range of ±50% of their initial values. On the other hand, all events for interest 4 are produced with the constant initial value.

Figure 3.10 shows the percentage changes of the collected values of interests compared to the last value observed for the interests. Figure 3.10 denotes that interest 4 has the highest fluctuation and interest 2 packets are produced with the least fluctuation.

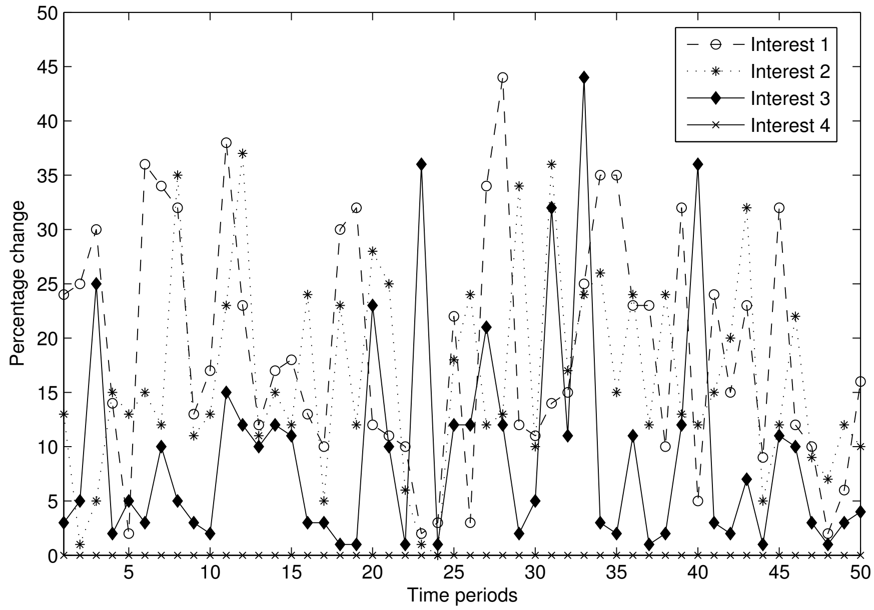


Figure 3.10: Percentage changes in the collected values for interests

Table 3.1 summarizes the rankings of interests and their percentages in the simulation. Figure 3.10 and Table 3.1 show that Interest 1 and 2 have highest percentage changes. 58% of the time Interest 1 and 32% of the time Interest 2 is the highest ranked interest. Since Interest 4 has constant values, it always has the lowest ranking.

Table 3.1: Interest ranks and percentages

Rank	Interest 1	Interest 2	Interest 3	Interest 4
1^{st}	58%	32%	10%	0%
2^{nd}	36%	52%	12%	0%
3^{rd}	6%	16%	78%	0%
4^{th}	0%	0%	0%	100%

Figure 3.11 shows the calculated weight values for each interest during simulation. Interest 1 has the highest ranking as the observed values for interest 1 has the highest changes. Although interest 4 has the lowest ranking throughout the simulation, its weight is not reduced to zero in the system. Additionally, there are spikes in the values observed for Interest 3, which has otherwise small fluctuations. However these spikes do not affect the ranking of the interest drastically, which shows the approach is not sensitive to abrupt changes.

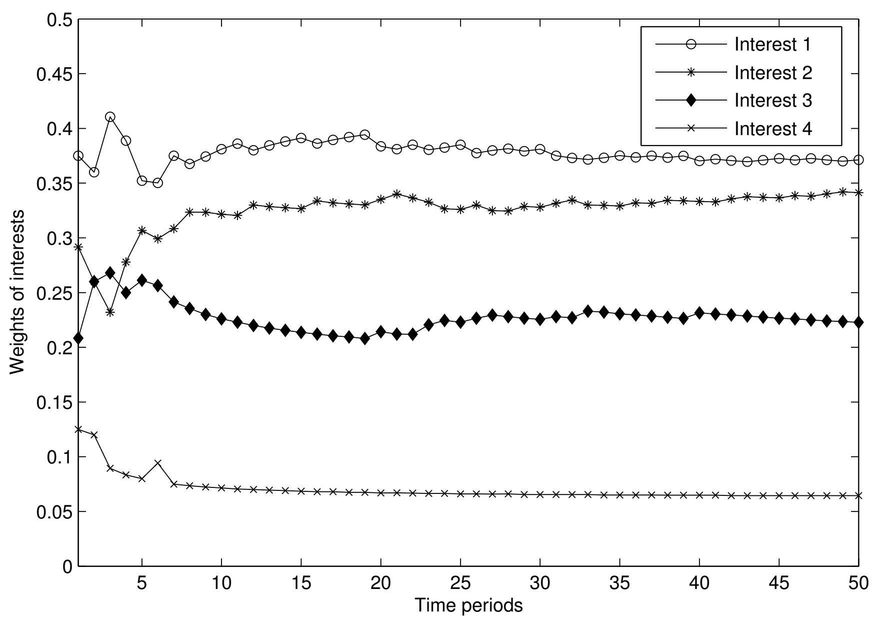


Figure 3.11: Weights of the interests

As a comparative analysis, the weight values of each interest are also calculated according to traditional sports ranking, which depends on the winning percentage. Figure 3.12 shows that this method results in zero weight for the worst ranked interest in the experiment, which prevents collecting information in response to that interest. Moreover the highest fluctuations in one time period (wins) are more affective compared to results in Figure 3.11, which provides a higher advantage for the winning interest, especially in the beginning of the simulation. While Interest 1 gains approximately a weight of 0.5 with this ranking method throughout the simulation, its average ranking is 0.38 in our protocol.

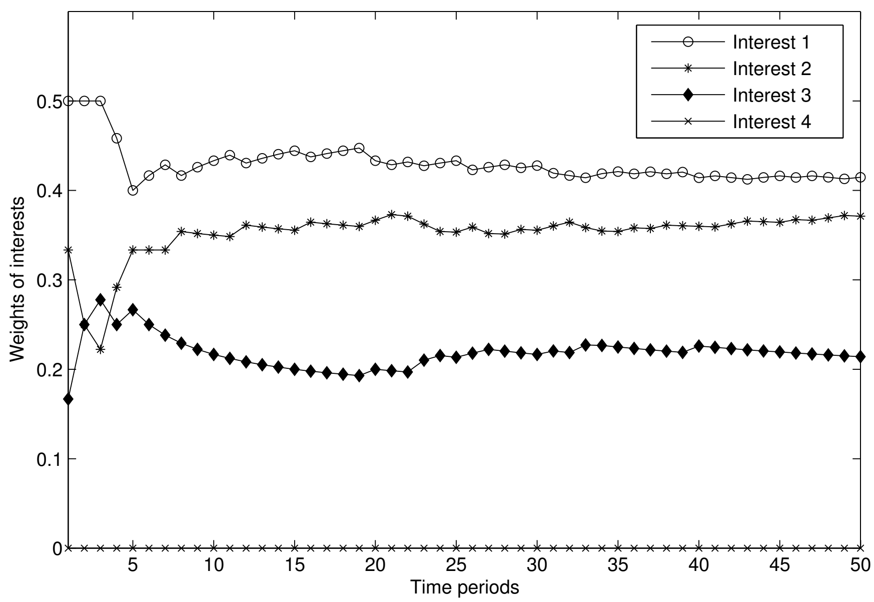


Figure 3.12: Weights of the interests with win-ratio ranking

The dynamic ranking improves the performance of the network when the interest values are updated according to the ranking. However frequent updates result in increased traffic load and energy loss. Therefore the defined l value to update the network is critical. Table

3.2 shows the required number of updates for different values of l. When l is set as 8x10^{-3} or higher, five updates are required in the actor's area. In other words, only five updates are required even if the actors are arranged to update the interests when sum of all changes in the weights of interests is 3% or more.

Table 3.2: l values and required number of updates

l ($\times 10^{-4}$)	288	250	225	200	175	75	63	50	38	25	13
Number of updates	1	2	3	4	5	6	7	8	11	21	33

In the next experiment, observed values for all interests fluctuate for first three time periods. Then, only the collected values for interest 1 continue this behavior while the rest of the observed values are stable for the remaining part of the simulation.

Figure 3.13 shows the calculated weight values for each interest during simulation. At the end of third time period, interests 1, 2, 3 and 4 are ranked respectively. Ranking value of Interest 2 is around 75% of the ranking value of Interest 1 whereas the ratio of Interests 4 and 3 ranking values is almost 50%. After the end of third time period, differences among the weights of interests with stable observed values start to decrease. At the end of the simulation they acquire very close values while preserving the same ranking order.

The behavior observed in Figure 3.13 shows that the main deciding factor of ranking interests is the change in the observed values as it is one of the critical goals of its usage. In addition, Interest 1 is not assigned with an extremely high weight although it is the only interest with changing observed values. Therefore the system keeps the effect of changes in the beginning of the simulation while giving more importance to fluctuating traffic type.

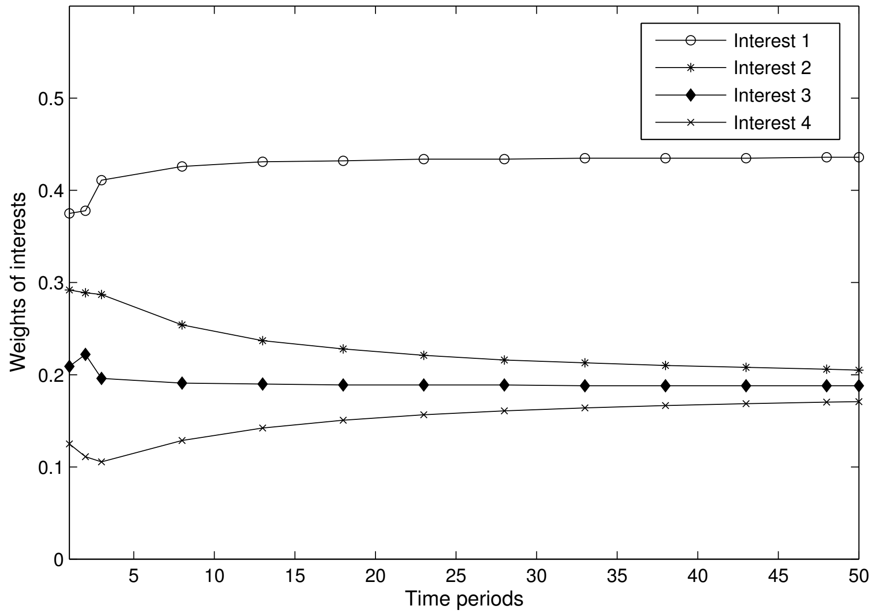


Figure 3.13: Weights of the interests

Figure 3.14 shows that Interest 1 becomes very dominant when traditional win-ratio ranking is used in the same conditions. Interest 2 and 3 lose half of their weights in 10 time periods and the worst ranked interest is assigned with zero weight. Therefore there is no protection for the traffic of interests other than Interest 1, which wouldn't be acceptable to collect periodical data even if the values do not change or slightly change.

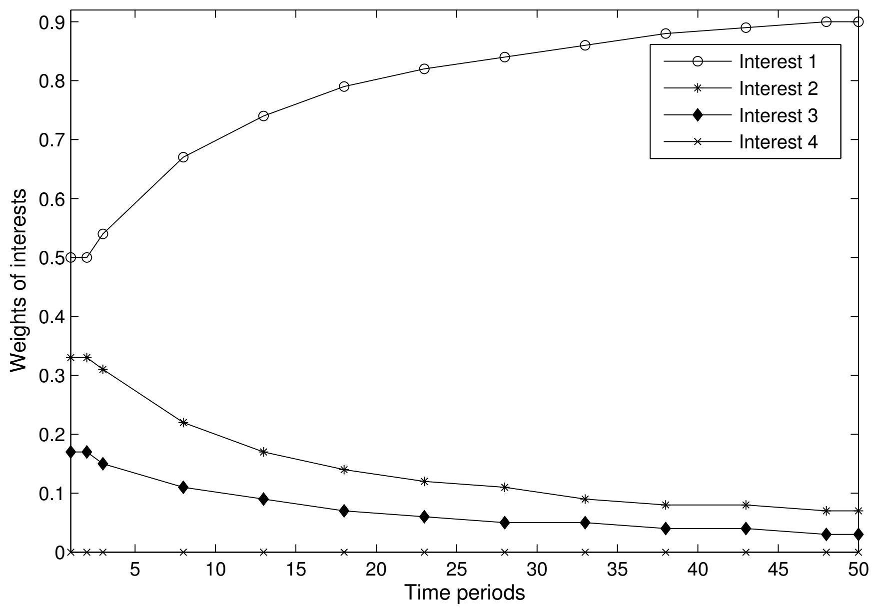


Figure 3.14: Weights of the interests with win-ratio rating

CHAPTER 4
SELF ORGANIZED ROUTING

In this chapter, we consider a WSAN application scenario in Amazon rain-forest with a river going through the forest (see Fig. 4.1). The actors are positioned at rare accessible parts of the area, while the sensor nodes are thrown in the river for data collection. Equipped with appropriate measurement technologies, sensor nodes are able to gather various kinds of data while floating in the river. For instance, Regan et al. [9] deployed such a multi-sensor system in the River Lee Co. Cork, Ireland to monitor water quality parameters such as pH, temperature, conductivity, turbidity and dissolved oxygen. Although nodes move basically in one direction in the river, they suffer from various peculiarities of the scenario such as permanent velocity changes, sudden stops by obstacles, etc.

In summary, the circumstances of the application scenario rise the following challenges for the design of an efficient routing protocol: (a) rapid changes of the neighborhood and actor association demands an efficient and reliable transmission of data from sensor nodes to the actors and (b) the dynamics of sensor nodes form a continuously varying topology requiring a highly adaptive network organization.

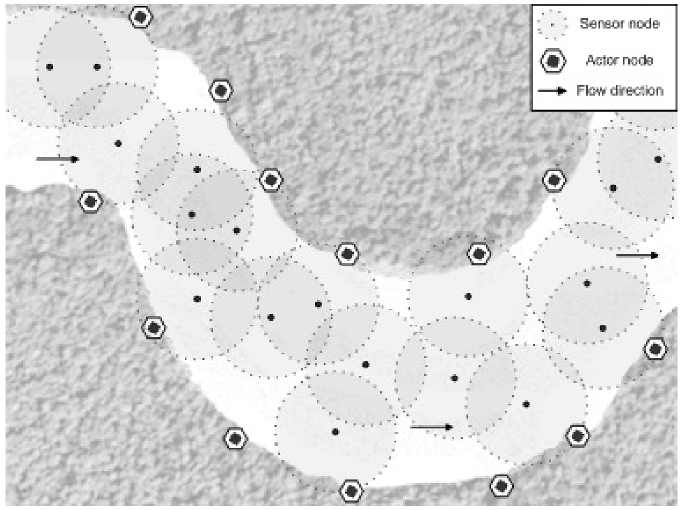


Figure 4.1: Amazon River application scenario.

In this chapter we propose Self-Organizing and Fair Routing Protocol (SOFROP) [15], [14] to address in particular the following critical issues of the Amazon scenario:

Indeterministic dynamics: When deployed in the river, the sensor nodes are subject to mobility. The sudden changes in current speed and direction combined with potential obstacles makes connectivity crucial among the nodes and the mobility pattern predictable only to a certain degree. Furthermore, the actor nodes can only be positioned on land and the sparsely accessible environment often impedes deploying actor nodes according to an ideal model, which would guarantee full connectivity at all times.

Restricted device deployment: Due to the node deployment restrictions the network structure has to allow multi-hop communication, i.e., sensors that are not directly connected to an actor should be allowed to communicate with other nodes to reach the actors. However, due to the dynamics of the river, routing paths continuously change and network re-organization occurs frequently. Therefore, the network organization should be done locally, avoiding superfluous message exchange, and it also must enable efficient realization of the routing protocol.

Effective data transmission: The QoS in WSANs is characterized according to the employed applications, each of which has various constraints such as reliability, latency and robustness. In SOFROP, we convey the QoS supporting characteristics of LRP-QS to Amazon river scenario, in which the efficient utilization of the available bandwidth and the minimization of packet drops are critical.

4.1 System model

The separation of the network organization from data transmission shows several benefits since the network organization phase adjusts the topology of the sensor nodes to enable efficient routing on the resulting overlay network. This separation reduces route failures and packet delay, while increasing the network throughput [159]. Hence, SOFROP is divided into two phases: the first phase is concerned with the network organization, where an overlay network is formed and continuously adapted. The second phase is responsible for the data transmission. In this section, the system model is explained briefly and the two phases of SOFROP are described in detail.

We consider a wireless actor and sensor network N with the number of nodes $|N| = n$. The wireless network N consists of a set of actor nodes A and a set of sensor nodes S, equipped with wireless communication capabilities. Our model also includes a sink node responsible for data aggregation and enabling connectivity to a backbone network. Each element in N is assigned to a transmission range r with a circular transmission area covering a total area of $r \cdot \pi^2$. The sensor nodes and actors in S are assumed to have maximum transmission ranges r_s and r_a, respectively, with circular transmission areas, where $r_s < r_a$ due to better computation and communication capabilities of the actors. For communication between two nodes, a bidirectional connection must be established, i.e., a device s_1 has to be in the transmission range of s_2, i.e $d(s_1, s_2) \leq r_s$.

4.1.1 Sensor nodes

For each sensor node s in S, we assume a neighboring list $Neigh(s) \subset N$, the set of nodes that are directly connected to s, such that $\forall u \in Neigh(s), d(s, u) \leq r_s$. $Neigh(s)$ is built initially when a node enters the network and updated with an update frequency f or triggered by an event.

Every node is able to communicate only with its current one-hop neighbors (a sensor node or an actor), thus all communication in this model is locality preserving. Geographical positions of the nodes are assumed to be unknown. Since data are transmitted in only one direction and only local information is used, no multi-hop control communication need to be applied. Communication links may fail or disappear from the network caused by obstacles for instance. Thus, the neighborhood of a node changes over time and nodes move with random and nonconstant speed, acceleration and directions.

4.1.2 Actors and the sink

SOFROP manages actor-actor communication efficiently in order to save battery lifetime. For this the actor nodes use their full transmission range in two cases only. One of these cases is the time when the network is initialized, in which the actor nodes and the sink create a network by using their full transmission ranges. In our particular application scenario, the actor nodes are positioned such that each one has at least one actor or sink in its transmission range. The sink communicates only with actors and it is also positioned in the transmission range of at least one actor. Otherwise the sink would be required to receive the collected data through the sensor nodes, which would create severe packet loss and delay conditions in the network. Considering this layout and the small number of actor nodes, the following steps are taken to form the links among actors and the sink:

- The sink starts the formation of links by flooding its ID and hop count (initialized as 1) encoded in a packet.
- This packet is forwarded in the network among actors and each actor saves the ID of the actor from which it received the packet with the lowest hop count as the destination for data traffic.

- The packet is retransmitted with an incremented hop count only if its hop count is less than the actor's.

The other case when an actor uses its full transmission range is when it has data to exchange, consolidate and transmit to the sink. Other than these two cases, actors use the same transmission range as the sensor nodes in the network organization phase and in communication with the sensor nodes. Although actor nodes typically have stronger resources and more energy budget relative to sensor nodes, resource constraints apply to both sensors and actor nodes [6]. Therefore this approach extends the lifetime of the actors, which is an energy-efficient feature of SOFROP. However, it is important to note that the actor-actor communication is not the main focus of SOFROP.

4.2 Self-organized and fair routing protocol

4.2.1 Network organization

The clustering is employed for the network organization in SOFROP. SOFROP must deal with the fact that actor nodes are pre-assigned cluster heads and are not supposed to change their status throughout the life-time of the network. Additionally, due to restrictions in the deployment of actor nodes, multi-hop clusters must be created as a remedy and the number of actor nodes must be minimized. The mobility of sensor nodes increases the number of re-affiliations to the actors.

4.2.1.1 SOFROP overlay network setting

The algorithm to create the overlay network does not require any initial configuration besides that each node must choose a value between 1 and k, its weight. The weight 0 is exclusively assigned to the actors. We assume that the only information available for a sensor node s is the information of the direct neighbors $Neigh(s)$ and their corresponding weights $w(Neigh(s_i))$. The beaconing is most commonly used to provide this information. However if beacon (or heartbeat) approach is used in the network, then the sensor nodes are required to transmit a packet periodically even when there is no neighbor node to receive this packet. Although beaconing is commonly used in sensor networks, it should be avoided when possible due to the energy constraints of sensor nodes. To address the energy requirements of the Amazon scenario, we propose a different approach to transfer the weight information:

- Only the actor nodes generate packets periodically, from the start of the network lifetime to the end. These packets are called *Area Configuration Packets* (ACP). An ACP includes actor ID and hop value fields. The actor initializes these fields respectively with its ID and hop value.

- A sensor node receiving the ACP drops the packet if the hop value on the packet is greater than or equal to its own hop value. Otherwise the node stores the values in actor ID and hop value fields of the packet and retransmits the packet with an incremented hop value.

- If a node loses its connection to the actors, it sets its hop value to the maximum hop value defined for the network. A node loses its connection when it doesn't receive an ACP (either directly from an actor or by retransmission of other nodes) for the predefined time defined for the network.

The nodes that lose the connection will be only in "listening" mode and they will not transmit any packets while actor nodes periodically send ACPs. This structure is suitable for WSANs since the complexity and resource requirement is focused on the actor nodes and it requires less energy than beaconing for the sensor nodes.

4.2.1.2 SOFROP clustering algorithm

The network structure in SOFROP is formed and maintained by the state transitioning rules of the clustering algorithm. Consider a node v with weight $w(v)$. The state transition for node v is given in Algorithm 3.

Algorithm 3 The state transitioning of a node v

$min.weight = w(v)$
for $i \in Neigh(v)$ **do**
    if $w(i) < min.weight$ **then**
        $min.weight = w(i)$
    end if
end for
if $min.weight < w(v)$ **then**
    $w(v) = min.weight + 1$
else if $w(v)! = k$ **then**
    $w(v) = w(v) + 1$
end if

As each node applies Algorithm 3, the network structure is formed by copying the lowest neighbor weight increased by one as the sensor nodes move into the transmission range of the actors. This property is important in creation of a hierarchical structure. When a lower weight node, which is not an actor attracts surrounding nodes with higher weights, this node successively increases its weight to avoid a fragmented structure.

Isolated nodes with weight k are physically able to communicate with neighbors having weight k, but according to our network organization phase no logical communication path is built. In order to include the isolated nodes in the network, a solution could be to increase the difference of 0 and k, forcing this phase to build longer communication paths. The network designer, however, must consider the velocity and perturbations of the river affecting the nodes. If the paths become extremely long, no effective routing can be conducted or the messages from the most distant node may fail to reach the actor node. For that reason, the difference between 0 and k must be chosen according to the environmental conditions. Thus, sensor nodes outside the coverage area of the actor nodes are simply ignored and they do not influence the remaining network due to their k-weight.

This phase uses only local information for the decision making process and all the nodes rapidly update their data as the network structure changes. In SOFROP, sensor nodes use only the information on the packets they receive; they do not keep any global data about the network.

4.2.2 Data transmission

The network organization phase provides sensor nodes the information about lower hop neighbors and the number of hops needed to reach the closest actor. The sensor nodes collect information from the environment as they float in the river with the objective to transfer data to the actor nodes.

"Interest" term is used in this protocol as in Chapter 3. Sensor nodes must be aware of the sink's interests in order to gather the required information from the environment while they float in the river. One possible approach to this problem is conveying the interest information to sensor nodes as they float in the river. This method allows a sink to dynamically distribute or change its interests anytime in the network. However in our application scenario, the sensor nodes are mobile and they move with the flow of the river, creating random and hard-to-estimate paths. The current of Amazon River can reach up to a speed of 7 km/hr. Having this speed and mobility pattern, some sensor nodes will be accessible for very short periods of time, which may not be enough to convey the interests and collect data from the environment according to these interests. Moreover, a sensor node may observe important events before it receives the interests, which can result in data loss. This may be critical for the network since the sensor nodes do not follow repeating paths. Changing interest distribution also requires a high number of updates as the actor areas continuously change in SOFROP. Therefore conveying the interests is feasible in a more stable scenario, for instance when the nodes are stationary. In SOFROP, the interests are predefined at sensor nodes before they are thrown into the river. Each sensor has a predefined list of the information to be collected from the environment, called the “interest table”.

As in Chapter 3, a node capturing an event encodes data packets with the rate it transmits them (α_p) and the interest (i_p) that the packets belong to. The receiving node takes the forwarding decision related to this packet based on the packet's information. We use eight bits to express the packet transmission rate and three bits to express the interest.

When an event is captured by a sensor node, the node checks its interest table to decide whether the sink needs to be notified of this event or not. If there is an interest for that event in the interest table, it is called an *on* interest for that node and the node generates data packets to report the event to the closest actor. A sensor node keeps the total number of *on* interests. The maximum packet transmission rate of a sensor node is called output capacity (C_o) in SOFROP. The rate field of the received packets is used to determine the remaining output capacity (C_r) of the node. C_o is reduced by the rate value on the first packet of an interest and recorded as the remaining output capacity (C_r) of the node.

Sensor nodes have buffers with predefined sizes. The buffer acts as a temporary space where the packets are held until the output link is available. Sensor buffers are simple in SOFROP; they work in first in first out (FIFO) fashion, outputting packets in the order they arrive. If we assume that the packet arrival process at each input link is a Bernoulli process with success probability p_s, the number of packet arrivals (A) at the buffer during a given time window has the binomial probability mass function and the probability generating function of the Bernoulli random variable with parameter p_s is as follows:

$$G_A(z) = \sum_{i=0}^{n} a_i z^i = (1 - p_s + p_s z)^n \tag{4.1}$$

A sensor node does not drop any packets when $C_r \geq 0$, which means the sensor node's resources are adequate to serve the received packets. When $C_r \geq 0$, the number of packets in the buffer at the end of the k^{th} time window (B_k) can be defined in terms of the number of packets in the buffer at the end of the $(k-1)^{th}$ time window and the number of packets arriving during the k^{th} time window (A_k) as follows:

$$B_k = max(0, B_{k-1} + A_k - 1) \tag{4.2}$$

The underlying stochastic process of B_k can be described by a Discrete Time Markov Chain (DTMC) with states $q_i = P(N = i)$ [160]. The state diagram of the DTMC is shown in Fig. 4.2.

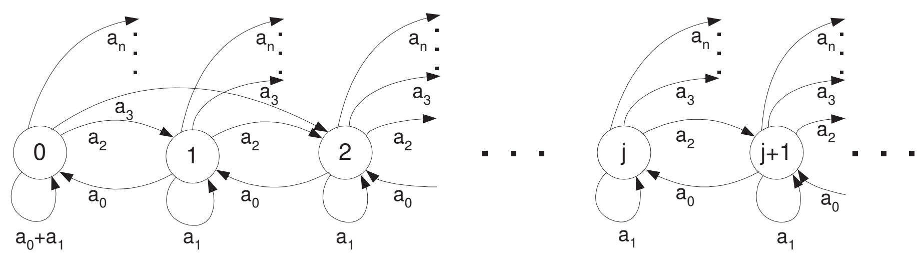


Figure 4.2: State diagram of the DTMC for the queue length of a sensor node.

If the sensor node does not drop any packets for a period of time, it means $np_s \leq 1$ in one time window of this period. Then the steady-state of the number of packets in the buffer exists. Consequently the buffer occupancy can be formulated as follows:

$$B_k = max(0, B + A - 1) \tag{4.3}$$

Then its probability generating function (pgf) is found as follows:

$$\begin{aligned}
G_B(z) &= \sum_{j=0}^{\infty} P(B = j) z^j \\
&= \sum_{j=0}^{\infty} q_j z^j \\
&= a_0 q_0 + \sum_{j=0}^{\infty} P(B + A - 1 = j) z^j \\
&= a_0 q_0 + \frac{G_B(z) G_A(z) - a_0 q_0}{z} \\
&= \frac{a_0 q_0 (z - 1)}{z - G_A(z)} \qquad (4.4)
\end{aligned}$$

The probability generating function satisfies $G_B(1) = 1$. Since $\lim_{z \to 1} a_0 q_0 (z - 1) = \lim_{z \to 1} z - G_A(z) = 0$, we can apply l'Hopital's rule:

$$1 = G_B(z) = \frac{a_0 q_0}{1 - G'_A(1)} = \frac{a_0 q_0}{1 - n p_s} \tag{4.5}$$

Therefore $a_0 q_0 = 1 - n p_s$. After substitution,

$$\begin{aligned}
G_B(z) &= \frac{(1 - n p_s)(1 - z)}{G_A(z) - z} \\
&= \frac{(1 - n p_s)(1 - z)}{(1 - p_s + p_s z)^n - z} \qquad (4.6)
\end{aligned}$$

The expected value of $G_B(z)$ is equal to the mean steady-state queue size of the buffer. It is found by differentiating $G_B(z)$ with respect to z and taking the limit as $z \to 1$:

$$E(B) = \frac{n(n-1)p_s^2}{2(1-np_s)} = \frac{(n-1)}{n}\frac{(np_s)^2}{2(1-np_s)} \tag{4.7}$$

The routing of the packets are handled as in Chapter 3, we define a fair rate (α_f) value to forward/drop packets. The fair rate for a node depends on the number of *on* interests (N_i) and defined as follows:

$$\alpha_f = C_o/N_i \tag{4.8}$$

It is important to note that if all the packets are received with rates greater than α_f, α_f is the maximum rate value that a packet will be encoded with when C_r is negative. When C_r is negative but the rate of the packet is smaller than α_f, the packet is forwarded without changing the values in its fields.

When a sensor node is not affiliated with an actor area, it waits for an area configuration packet (ACP). The node does not transmit any packets while waiting for an ACP, but fills its buffer with the packets it generated according to the sensed events. This is an efficient and feasible approach since the topology of the network changes continuously and the node can transmit the generated packets whenever it receives an ACP from a sensor node. If the node cannot be affiliated to an actor area before its buffer becomes full, it keeps its buffer updated with the latest observations.

4.2.3 Illustration of SOFROP

An example sequence for network organization of four sensor nodes and one actor is demonstrated in Fig. 5.1. At the beginning of the sequence, the actor node has a weight of zero and all sensor nodes are initialized with the weight value k. The remaining sequence shows how the network structure is formed. When the sensor nodes get into transmission range of an actor, they start to take weights according to Algorithm 3.

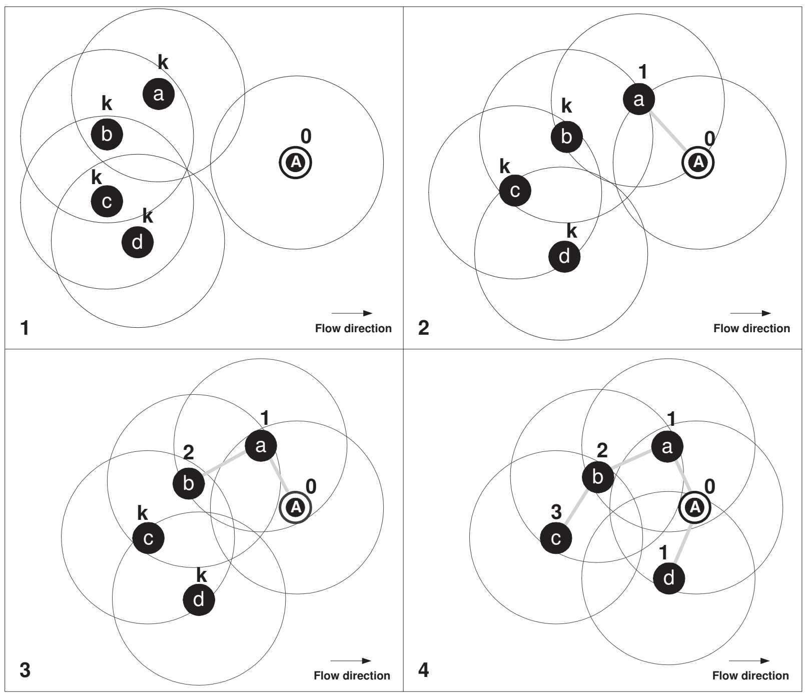


Figure 4.3: An example sequence of a small group of sensor nodes and an actor.

An example of a lower weight node attracting surrounding nodes is illustrated in Fig. 4.4, where one of the nodes is moving faster compared to the other nodes in the scenario. This is a possible case due to potential obstacles and unpredictable flow rate changes in the river. The fast-moving node initially has a weight of one since it is directly connected to the actor at the beginning, but it loses its connection to the actor after it moves further away. However it still receives ACPs since it is in the transmission range of a node with weight three. Then, it increases its weight to connect to the closest actor.

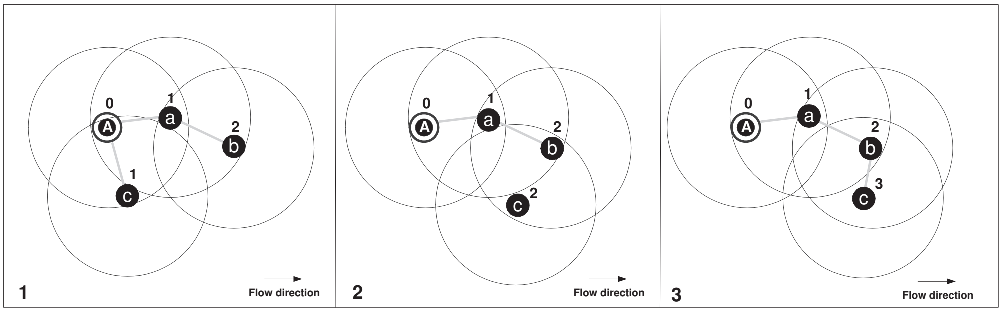


Figure 4.4: An example sequence on avoidance of a fragmented structure.

A sequence of the dynamic overlay network produced by network organization is denoted in Fig. 4.5. Three actors are deployed uniformly at random and remain static while 60 sensor nodes are flowing from left to right, where the maximum hop-count allowed by the network organization is four. Note that Fig. 4.5 depicts only one of the different outcomes possible due to asynchrony.

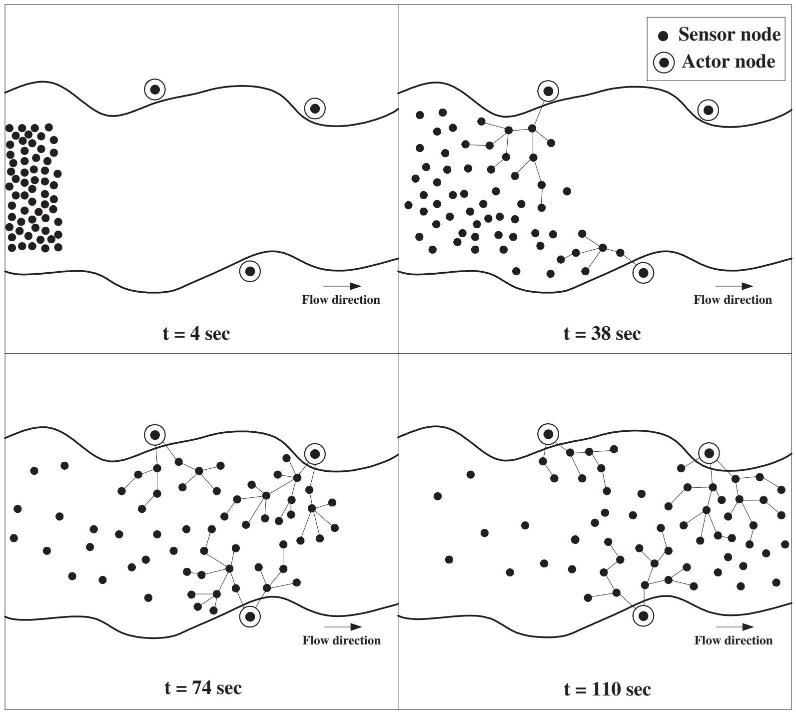


Figure 4.5: A sequence of dynamic overlay network formation.

An example sequence of packet transmissions in a system of four sensor nodes is shown in Fig. 4.6, 4.7 and 4.8. This system can be considered as a collection of any four sensor nodes in a network where SOFROP is employed. The initial state of the system is shown in Fig. 4.6. Nodes a and b have the hop-counts of three, node c and node d have hop-counts of two and one respectively. All sensor nodes are assumed to have an output capacity of ten packets per second. The packet type is presented in the packet header (T1: type-1 and T2: type-2). The value of the packet's rate field is shown with the numerical value and the destination of the packet is presented as a letter, showing the destination sensor node.

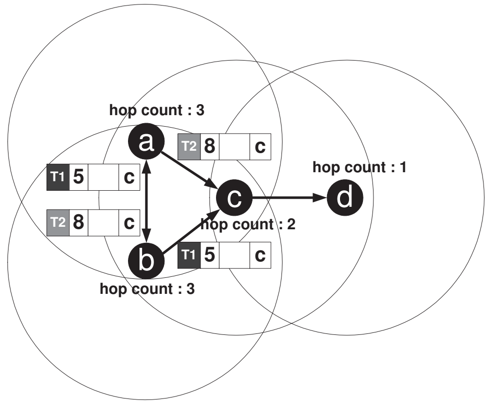


Figure 4.6: An example system for demonstration of routing and labeling.

Sensor node a has a packet with the rate field value of eight and sensor node b has a packet with the rate field value of five. Both of these packets are destined for node c and they belong to different interests. An important property of SOFROP is denoted in Fig. 4.7. Since b is in transmission range of a, it receives the packet from a. However the packet is not processed and directly ignored at b since the destinations for data at sensor nodes are determined during the network organization. The interest table of node c is also shown in Fig. 4.7. The types of packets from a and b have the same priority and these priority levels are predefined in the table of node c.

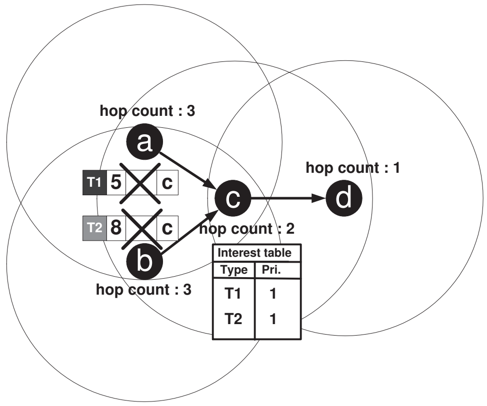


Figure 4.7: An example system for demonstration of routing and labeling.

The node c receives the packets of two different types and these two flows share the output capacity of node c. Since both of the interests have the same priority, they must share the output capacity equally according to SOFROP. Since the output capacity of the node is ten packets per second, the fair share of the bandwidth for two flows is five packets per second. The node c drops packets from the flow, which are received from node a. Therefore the rate values on the packets, which are received from a and forwarded to d, are updated according to the calculated P_d and α_p. In Fig. 4.8, the rate field of the forwarded packet, which was received from a, is assumed to be changed to five, which would be the case in ideal conditions.

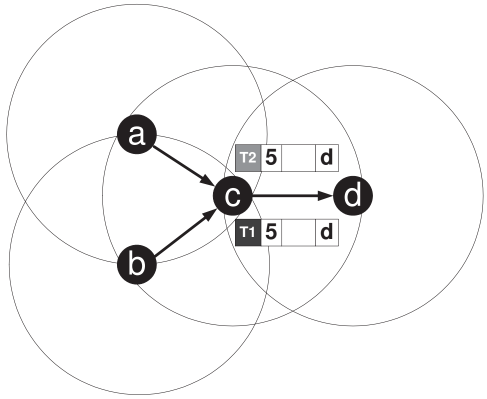


Figure 4.8: An example system for demonstration of routing and labeling.

In Fig. 4.9 and 4.10, the same system of nodes as in Fig. 4.6 is used with different values for the rate fields of the packets. The values of the rate fields are changed to six and two in this example for the packets coming from nodes a and b respectively. All packets have the same priority, which is predefined in the interest table of node c.

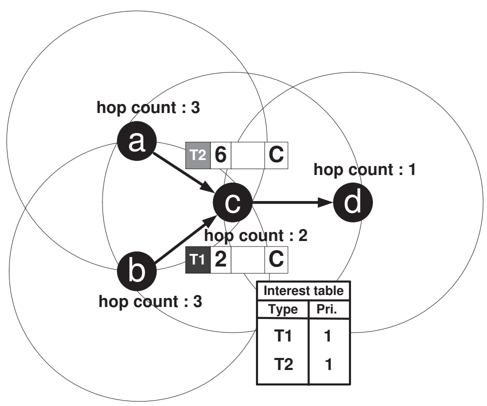


Figure 4.9: An example system demonstrating bandwidth utilization.

According to the initial property of SOFROP, which is described in Fig. 4.6 to 4.8, flows with equal priority share the output capacity of the transmitting node equally. Therefore if only that rule is employed by SOFROP, then the type-1 packets transmitted by node c will have rate values of two. This means most of the type-1 packets will be dropped according to P_d calculation although a big portion of output capacity of the node is not utilized. This is not acceptable in Amazon scenario where QoS has high priority as explained in the initial sections. Therefore in this scenario, packets of both flows are transmitted with unmodified rate fields.

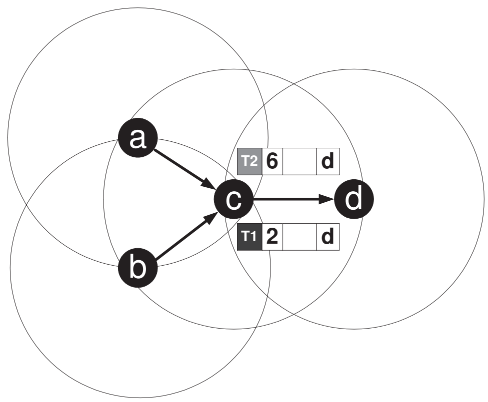


Figure 4.10: Both of the packets are forwarded unmodified.

In Fig. 4.11 and 4.12, the same system of nodes as in Fig. 4.6 is used with different rate fields of the packets to demonstrate that both packet priority and bandwidth utilization have important effects in the decision making process of SOFROP. In this example, the rate fields of the packets are changed to five and six. Different from the previous examples, the packets received from nodes a and b have different priorities. The priority of the packets from node b is twice the priority of the packets from node a.

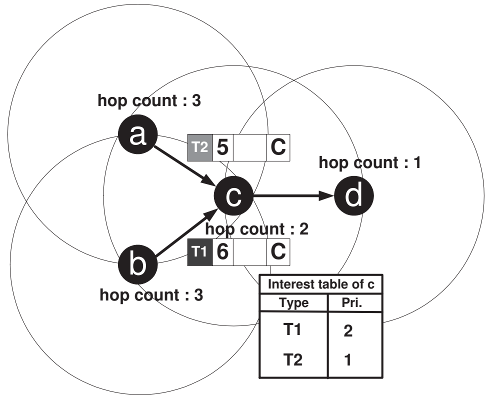


Figure 4.11: An example system for demonstration of routing and labeling.

The priorities of the flows are also important in this example when dropping the packets. Since there are two active flows on the node and the priority of one is half of the other, there will not be any packet drop from the flow with higher priority unless α_p becomes greater than $C_o \cdot 2/3$ according to SOFROP. Therefore packets from node b are forwarded without modification whereas the packets from node a are dropped with the corresponding P_d and the values of the rate fields of forwarded packets from node a are changed accordingly. In Fig. 4.12, the rate field of the forwarded packet, which was received from node a, is changed to four.

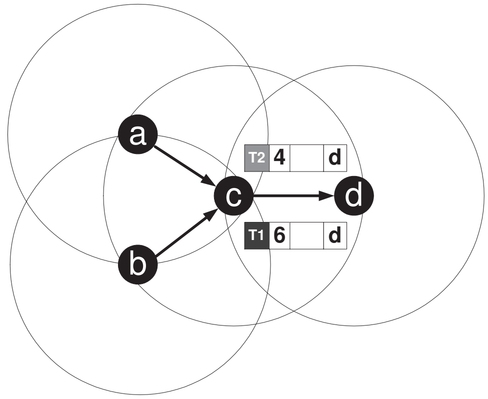


Figure 4.12: An example system for demonstration of routing and labeling.

4.3 Simulation study

4.3.1 Simulation environment and metrics

The simulations are carried out by OPNET modeler [158]. The transmission range of a sensor node is 40 meters, a realistic range for a sensor node (for instance Cerpa et al. [161] finds the transmission range of second generation Mica-2 motes to be between 20 and 50 meters in an outdoor habitat). The assumptions include a queue size of 20 packets and a data rate of 10 packets per second. The IEEE 802.11 is used as the underlying MAC layer of the nodes and wireless LAN model in OPNET allows transmission power of a node to be defined as an attribute by means of OPNET's transceiver pipeline implementation. The relation between the transmission power of a node (T in Watts) and its transmission range (r) is defined as $T = \left(\frac{4\pi r}{0.12476}\right)^2 \cdot 10^{-12.5}$. Table 6.4 summarizes the simulation parameters used in our experimental setup.

Table 4.1: Simulation parameters

Number of sensor nodes	60
Total area	200x300
Sensor transmission range	40 (meters)
Number of actor nodes	4
Traffic type	CBR
Data packet size	256 bytes

The communication graph is built according to the SOFROP system model. In each simulation, a network topology is generated with the sink located at one side of the area and actor located randomly either on the sides of the river or on the islands. The communication links may fail or disappear from the network caused by several reasons such as obstacles in the river. A random mobility profile is created in OPNET modeler for the sensor nodes so that the nodes are moving in the watercourse with the settings given in Table 4.2.

Table 4.2: Mobility Settings

Starting point	x= 0-10 m; y = 0-300 m
Destination point	x= 100-200 m; y = 0-300 m
Pausing time	0-10 sec
Speed	0-3 m/sec

The protocol stack of the sensor node model is created in OPNET modeler as shown in Fig. 4.13. Wireless local area network (WLAN) receiver and transmitter form the physical layer of a sensor node model. WLAN MAC layer interface is the data link layer used in OPNET 802.11 implementations and it is the interface between the routing layer and the

WLAN MAC layer. The attributes of underlying IEEE 802.11 MAC layer used are shown in Table 4.3. The routing layer is where algorithm of SOFROP is mainly implemented. Sensor sink and sensor source modules serve as an application layer for the sensor node model. The source module is capable of creating packets when the sensor node is required to transmit information about its conditions. The sink module is capable of generating responses for the queries to the sensor node. It can also be used to collect necessary statistics.

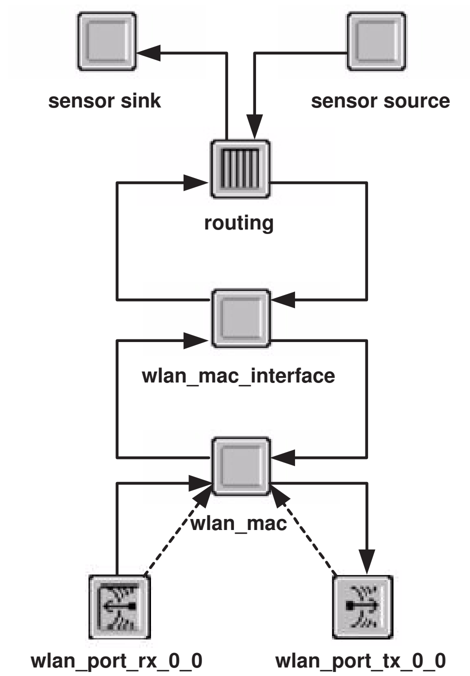


Figure 4.13: Sensor node model created in OPNET.

Table 4.3: MAC layer attributes of sensor nodes

Physical characteristics	Direct sequence
Transmit power (W)	$8.02 \cdot 10^{-6}$
Reception power threshold (dBm)	-95
Channel settings	Auto assigned
Short retry limit	7
Long retry limit	4
PCF	Disabled
HCF	Not supported

Amazon River is the second largest river in the world with islands on it and its width ranges from a few hundred meters to 10 kilometers even at low season. The simulation study does not reflect actual dimensions of the Amazon scenario since we concentrate on the reproducibility of the results in the current work. As a part of the future work, we plan to conduct simulations with exact dimensions and more complex mobility models, and real world experiments on site.

4.3.2 Simulation results

We study the effect of the proposed algorithm with the following simulation metrics: fairness, number of packets received, maximum hop value in the network and number of sensor and actor nodes.

4.3.2.1 Experiment 1

In order to create data traffic, twenty traffic sources are randomly placed in the network. These event sources simulate the points where it is possible to make observation for the sensor nodes in the network. They generate three different types of packets with constant rate of 10 packets per second, which creates congestion and bottlenecks in the network from time to time depending on the dynamic topology. All three of these traffic types have equal priorities. While 50% of the produced packets are type-1, type-2 and type-3 traffic have 30% and 20% allocations respectively. Twenty simulation runs were executed; the actor nodes and event sources are distributed randomly in the area for each simulation run. In order to see the effect of fairness, the same set of simulations is performed without using the fairness property of SOFROP. In other words, the rates or priority fields of the packets are not taken into account while taking routing decisions.

In Fig. 4.14, we can observe two important characteristics of SOFROP. First, the number of received packets for each type is very close to each other. Since packets from each type are produced at very large numbers, they create bottlenecks in the network. At these bottleneck points, higher the rate on a packet, greater the chance of that packet being dropped according to the routing principles of SOFROP. SOFROP drops more packets from the type of traffic with higher rate among the types with same priorities in a congestion situation. This is a desired property for the network since the sensor nodes are collecting information from the network and information on a single traffic type should not suppress the others. However when we take fairness properties out of SOFROP, we cannot observe the same property. Type-1 traffic receives more resources than for the other types in this case and additionally the total number of received packets is smaller.

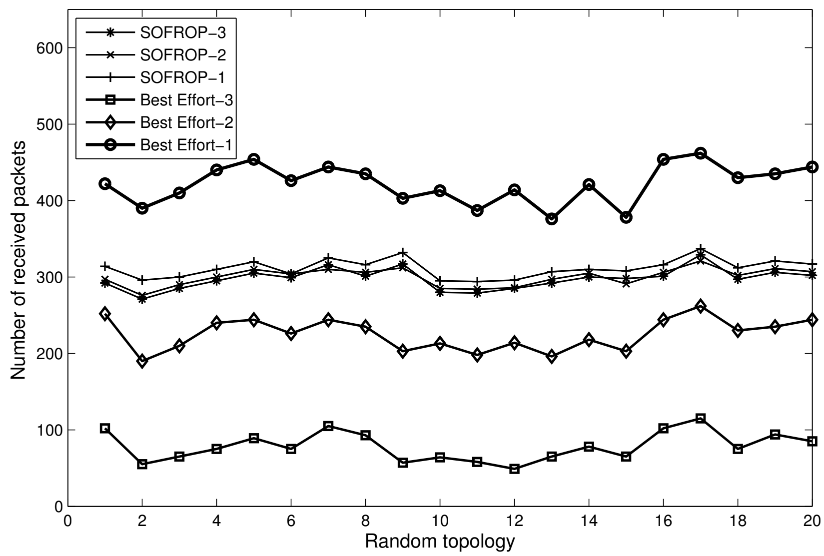


Figure 4.14: Number of packets received by actors in Experiment 1.

4.3.2.2 Experiment 2

In order to test another property of SOFROP, the same set of simulations is run with different settings. The first setting change is in the percentages of the produced traffic types. In this experiment, 50 percent of the produced packets are type-1, 45 percent of them are type-2 and only 5 percent are type-3. As for the second change, we also include priorities in this case. While type-1 and type-2 packets have the same priority, the priority of type-3 packets is three times larger. This means that type-3 information is critical for the network. Fig.

4.15 shows the number of packets of each type received by the actors. The results show that SOFROP protects the critical type of traffic and drops a very small number of packets.

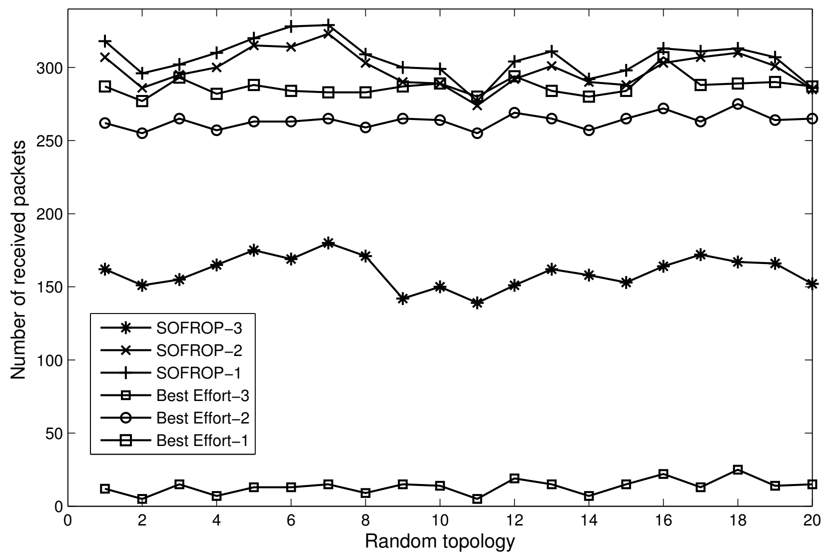


Figure 4.15: Number of packets received by actors in Experiment 2.

4.3.2.3 Experiment 3

Another set of twenty simulations with the same settings in Experiment 1 is run without using the bandwidth utilization property of SOFROP. In Fig. 4.16, the lines corresponding to the runs with SOFROP are labeled as "SOFROP" and the lines corresponding to the runs without the utilization property are labeled as "No BW Util.". Therefore in these experiments, the only constraint is fairness but the utilization of the resources is not taken into account while taking routing decisions for "No Util." cases.

In Fig. 4.16 the number of received packets for each type of packets is very close to each other in both cases. This is the property observed in Fig. 4.14, which is also expected in the runs without utilization since the only constraint is fairness. However we also observe that the number of received packets by actors without utilization property is less than SOFROP. The output capacity of each sensor node in the network is used at most three times the rate of the flow with the minimum rate since all flows have same priorities.

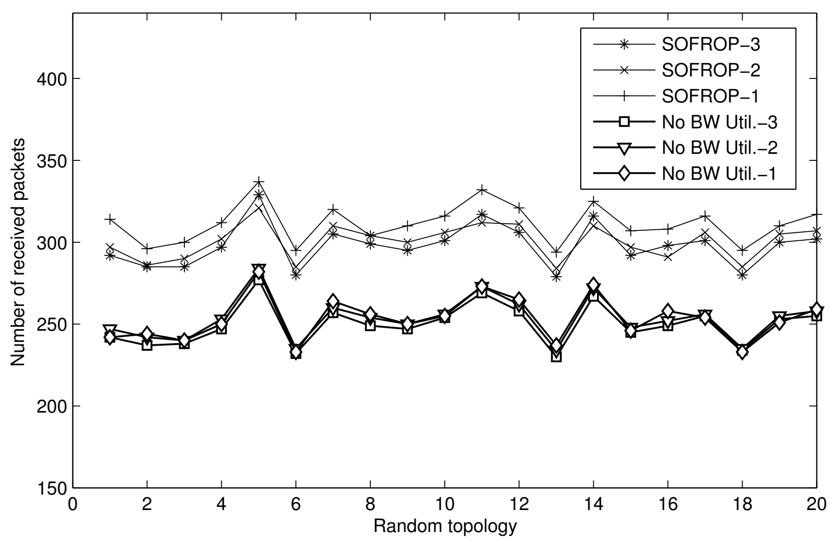


Figure 4.16: Number of packets received by actors in Experiment 3.

4.3.2.4 Experiment 4

The delay characteristics of SOFROP are observed by using a simulation set similar to the one in Experiment 2. Delay values depend fairly on topology in our application scenario since the path of a packet changes with the topology and the number of actors. It is shown in Fig. 4.17 that SOFROP performs clearly better when it is fair, which is critical when combined with the previous results. The results indicate that SOFROP not only protects critical packets but also delivers packets with a low average delay, which is another main QoS parameter.

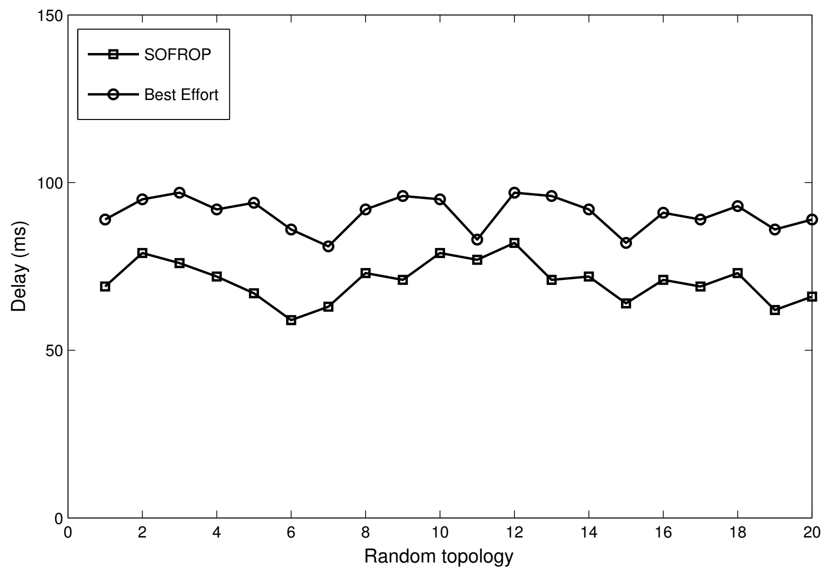


Figure 4.17: End-to-end delay in Experiment 4.

4.3.2.5 Experiment 5

The SOFROP's coverage properties are investigated using the same simulation settings as the previous experiment and the number of connected and unconnected nodes is observed in this experiment. Besides we measured the hop distribution for k values in between 3 and 6. The total of 25 simulations are run for each value of k, where each simulation period ends as the first sensor node moves out of the area. The average numbers of sensor nodes with different hop-count values are presented in Fig. 4.18 for each k value. The results show that number of unconnected nodes decreases by 20 to 30% as k is incremented by 1. The number of nodes associated with an actor increases with increasing k; for example the average number of unconnected nodes is 20 when $k = 5$. Fig. 4.18 also shows that at least 45% of the nodes are in 2-hops distance for all values of k. Along with the other simulations, this experiment also denotes high adaptability of SOFROP's network organization to mobility.

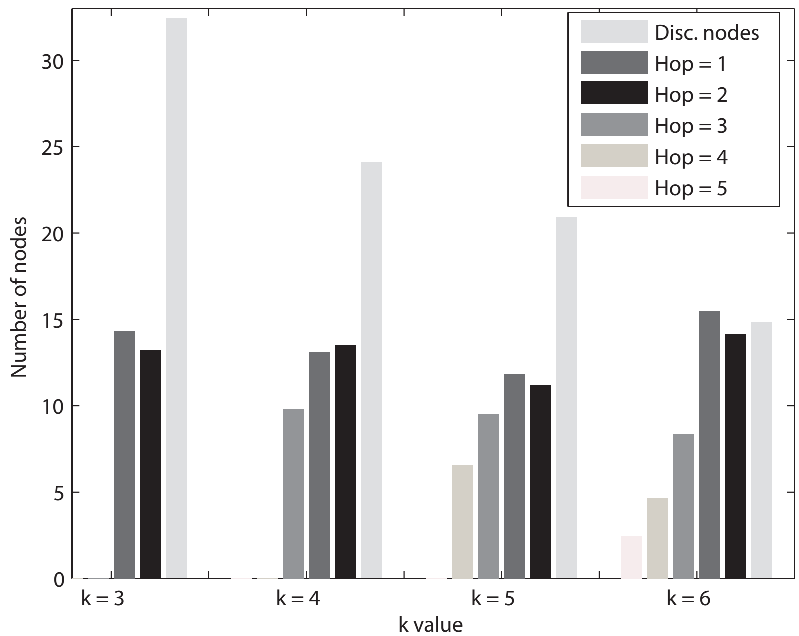


Figure 4.18: Number of sensor nodes and their hop values for different k values.

CHAPTER 5
MULTI HOP LOCALIZATION

In Amazon rain forest scenario presented in Chapter 4, the sensor nodes collect information about the environment while they move in the river and transmit the collected data to the actors. The positioning process is critical in this environment to match the collected data with the position of a sensor node at a given time of the observation. In many application scenarios, the collected data can become unusable if not associated with the position and the time.

The river scenario introduces particular challenges for localization such as a continuous change of the communication topology. Interference techniques and time stamps are used in order to overcome these challenges and to reveal the paths that the sensor nodes follow through the river. Hence the main goal of the positioning algorithm is not creating a self-awareness of locations at sensor nodes but enriching the collected data with positioning information.

In addition to the specific locations, the paths of the nodes also deliver important information about the nodes and the environment such as speed, direction or the structure of the terrain. For instance, the detection of a significant change in the speed of a majority of the nodes in a certain location of the river may indicate that either the terrain is more inclined in that area or there exists a small waterfall. Another example would be that if at a certain point of the river, the differences among the speeds of the nodes are large, some nodes may have run into obstacles while the others flow through the river without any problems.

The main contribution of this chapter is the multi-hop locality preserving localization algorithm design, which enriches the collected environmental data with position information [16–18]. For the creation of the hierarchical network structure of sensor nodes and actors, a network organization algorithm is also designed and used along with lateration for localization. The organization in the network is dynamically adapted with the continuous change of the communication topology, turning mobility into an advantage for energy efficiency by eliminating the requirement for extra control messages. The accuracy of the event localization is improved by including path estimation in the algorithm.

The second contribution of this chapter includes the adaptation of a realistic mobility model according to the requirements of the river scenario. Since the sensor nodes drift in the river with the force of the current and the use of a realistic current mobility model is critical to analyze the accuracy of the localization algorithm. Amazon river has a high water load and low slopes over the river basin, resulting in a meandering structure (see [162]). Therefore, a subsurface meandering current mobility model with random surface motion is incorporated into our algorithm taking the requirements of our scenario into consideration. To the best of our knowledge, this is the first example of using meandering current mobility model with random surface motion for the simulation of sensor and actor networks operating on the surface of a river.

Additionally, a basic directional mobility model is also created without considering the any specific characteristics of the river. This basic model is used to analyze the effects of the realistic mobility on the performance of the algorithm.

The system is not intended to detect a specific event (e.g. fire or an intruder); instead it is used to monitor the river for a period of time and to create a detailed set of data for a complete analysis of the river. Therefore, different than most of the existing localization algorithms, the sensor nodes are not informed about their positions and no computation is specifically required at the sensor nodes by the positioning algorithm.

5.1 Network organization

5.1.1 Network layout

We consider a wireless sensor and actor network N with the number of nodes $|N| = n$. N consists of a set of actor nodes A and a set of sensor nodes S. Our model also includes a sink node responsible for data aggregation and connection of the environmental monitoring network to a backbone network. The sensor nodes and actors are assumed to have maximum transmission ranges r_s and r_a, respectively, with circular transmission areas, where $r_s < r_a$ due to better computation and communication capabilities of the actors. Every node is able to communicate only with its current one-hop neighbors, forming a locality preserving communication system.

Sensor nodes in the application scenario are deployed without any positioning adaptors. Positioning devices such as GPS receivers are avoided to enable longer network lifetime with the limited resources of sensor nodes. Besides long life-time requirements, the thick forest structure of Amazon is also a disadvantage for satellite communication. Therefore obtaining location information by GPS is not a viable option in Amazon river monitoring scenario. In our system, sensor nodes do not keep position information and each sensor node directly communicates only with its immediate neighboring nodes. The local information of a sensor node is forwarded to the actors via intermediate nodes. The actors acquire their positions either from an external source or the position information is encoded in the deployment phase.

Actors are positioned on the coastline or on the islands of the river such that each one has at least one actor in its transmission range. The selection of actor positions and the actor network affects the performance of the localization approach. However, the main focus of this work is localization; therefore, actor positioning is not analyzed in detail. Actors use their full transmission ranges only when communicating with other actors. When communicating with sensor nodes, the actor nodes use the same transmission range as the sensor nodes in order to have a bidirectional connection and save energy. The network among the actors serves as the layout network for the processing of the collected data. This layout network is formed by selectively flooding the network formation packet. Each actor sets the neighboring actor in which the first network formation packet is received from, as the destination for the collected monitoring data. Then, an actor can receive additional network formation packets. The actors, from which these packets are received from, are saved in a list to be used in cases such as a change in the communication backbone.

5.1.2 Weight assignment for nodes

The main goal of our location estimation approach is to enhance the environmental monitoring information collected from the river and create a detailed set of data with location and time information for a complete analysis of the river characteristics. Therefore, in contrast to most of the existing localization algorithms, the sensor nodes are not informed about their positions. This way, no computation is specifically required by the localization algorithm at the sensor nodes.

The mechanism for actor affiliation utilizes the weight values of the nodes. Each actor is assigned to a constant weight k, which is initialized with a value based on the characteristics of the network such as the communication ranges of the nodes or the physical constraints of the environment. Sensor nodes store weight values for the actors they are affiliated with. There is no initial configuration on the sensor nodes to affiliate them with the actors. The only data available for a sensor node s as it floats in the river are the direct neighbors $Neigh(s)$ and their corresponding weights, $w(Neigh(s_i))$.

The list of the weights for the actors, which a sensor is affiliated with, is stored in a weight table. The sensor nodes initially take random weight values between 0 and $k-1$. The weight value of each affiliation is updated according to "hop distance", h, of the node to the actor in which it gets updates from. Therefore the weight $w_a(v)$ of a sensor node v for an actor a is as follows:

$$w_a(v) = k_a - h_a(v) \tag{5.1}$$

The actors exchange packets with the sensor nodes in their transmission ranges. Each actor encodes the transmitted packets with its ID and weight k, denoting that the packet is originated at an actor. Each sensor node updates the weight values in its record via local updates. Thus the information about each actor is distributed and updated at the affiliation area of that actor.

A sensor node is capable of being affiliated with multiple actors and keeps the maximum weight for each actor it receives the packets from, as depicted in Fig. 5.1. The structure of the network is created as the sensor node continuously adapts its weight according to its local neighborhood. For each of its affiliated actor nodes, a sensor node's weight depends on the highest neighbor weight, M for that actor. The sensor node is assigned with the weight value of $M - 1$ unless it already has the same weight value. Hereby a hierarchical structure is dynamically formed and updated by creating an ordered tree structure. Therefore a sensor retransmits a packet received from an actor only if the weight value is less than the sensor node's weight for that actor. Otherwise the sensor node drops the packet to avoid unnecessary traffic and energy consumption in the network.

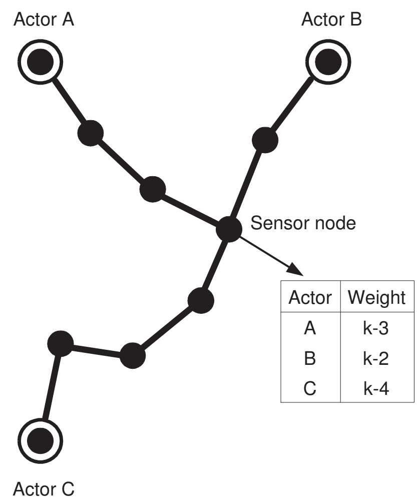


Figure 5.1: Single sensor node affiliated with multiple actors.

The condition, in which a sensor node doesn't receive any weight updates, is defined as the "loss of connection" for the sensor node. In that case, the sensor node sets its hop value to the minimum value defined for the network. Then it operates only in "listening" mode and does not transmit any packets. Listening nodes are physically able to exchange packets with neighbors having zero weight. However, there is no communication among these nodes.

Algorithm 4 describes the state transitions for the node v depending on its actor affiliation. The weight w of each affiliation corresponds to "k-hop distance" of a node v to an actor a.

The hierarchical structure of the network is created with the step, by which a sensor node continuously adapts its weight to its local neighborhood. For each of its affiliated actor node, a sensor node changes its weight to "the highest neighbor weight – 1" unless it already has the same weight value. Hereby a hierarchical structure is dynamically formed and updated.

Algorithm 4 The state transitions of a node v.

$w_a(v)$: The weight of node v for actor a
$\max(w(Neigh(v_a)))$: M
if v is a sensor node **then**
    if v is not affiliated with an actor node **then**
        $w_a(v) = 0$
    else if $M = k$ **then**
        $w_a(v) = k - 1$
    else if $M! = k$ & $M > w_a(v)$ **then**
        $w_a(v) = M - 1$
    else if $M < w_a(v)$ **then**
        $w_a(v) = w_a(v) - 1$
    else if $Neigh(v_a) =$ Empty **then**
        $w_a(v) = 0$
    end if
else
    $w_a(v) = k$
end if

When floating in the river, some sensor node groups can lose connection to the actors and particular nodes in the group can have weights higher than all of their neighbors without having any affiliations to the actors. According to the hierarchical clustering weight, these nodes attract surrounding sensor nodes with lower weights, which would cause unnecessary data traffic since nodes with lower weights would transmit monitoring data towards the nodes with higher weights. This is prevented in the algorithm by successively reducing the weight of a node when it has the highest weight among its neighbors. The weight adaptation continues until connecting to an existing actor or until the weight of the node becomes the minimum value for the network.

Fig. 5.2 shows an example of a sensor node's weight adaptation according to the algorithm described above. All the other nodes in this example are assumed to be stationary and k value for the network is set to four. The mobile node in Fig. 5.2 is initially not affiliated to any actors, so its weight is zero. Then the node becomes directly connected to an actor and its weight becomes $k-1$. As the node moves away from the actor node and affiliates with another actor node, its weight changes according to "the highest neighbor weight – 1" rule. Finally, it ends up not being affiliated with any actor node, its weight is decreased until it becomes zero and stays there as stated in the last step.

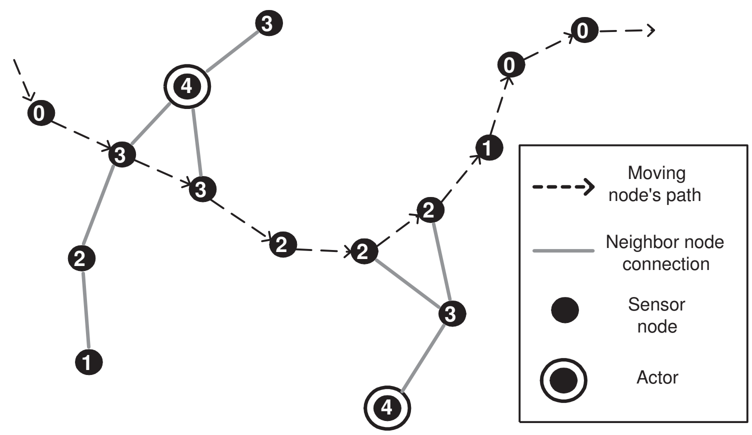


Figure 5.2: Weight adaptation of a sensor node.

Fig. 5.3 shows another example of a group of listening sensor nodes and their weight (depicted as w in Fig. 5.3) adaptation. All the nodes in this example are assumed to preserve their hop-distance to their neighbors during the specified period of time and k value for the network is set to five. The situation in Fig. 5.3 can be considered as the time when this group of four nodes just lost connection to an actor. Therefore none of the nodes is affiliated with an actor. As can be seen in Fig. 5.3 the weight adaptation terminates at sixth step when all the nodes have $w = 0$. In a realistic river environment, the neighbors of the nodes change continuously with mobility.

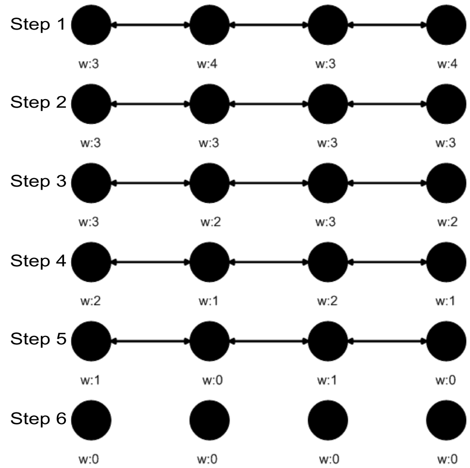


Figure 5.3: Weight adaptation for a group of sensor nodes.

5.1.3 Adapting k

One of the characteristics of our algorithm is that the nodes use only local information. The sensor nodes update their data locally as the network topology changes due to node mobility. The sensor nodes affiliated with an actor form the "affiliation area" of that actor. In other words, if the neighbors of a sensor node which are affiliated with an actor, have zero weights, then that sensor node cannot be affiliated with that actor and it is out of the coverage area. When a sensor node is outside of all the affiliation areas, it is called an "unconnected node".

The selection of k value of the actors is critical for the traffic and the energy consumption in the network. In our algorithm, actor nodes adapt their k values according to their observations. In order to find k, each actor follows the following steps:

- The k is initialized with a high value (depending on the size of the network area) at the actors. If the k value of an actor changes, the actor announces this change to the other actors via the backbone.

- As the sensor nodes float into the reception range of actor nodes, they start to form and update their weight tables as shown in Fig. 5.4. Actors also keep weight values for the other actors with the same method.

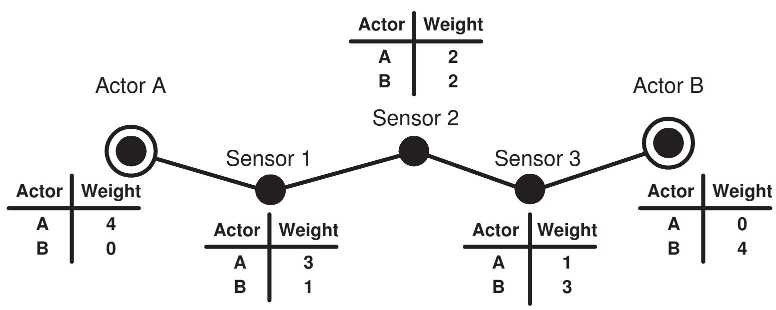


Figure 5.4: The weight tables of sensor nodes affiliated with two actors.

- Each actor computes a k_u value for all of its neighbor actors as follows:

$$k_u = \frac{k_n - w_n}{2} \tag{5.2}$$

where k_n is the neighbor's k value and w_n is the weight value of the actor for that neighbor actor. k_u represents the minimum number of hops between two actors. The k_u values are updated continuously such that the minimum is kept in the records for each neighbor actor.

- The distances from the neighbor actors are calculated. Actors have their location information, so the distance (d) between two actors, $i = 1, 2$, is calculated simply by using the Pythagoras theorem:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = d^2 \tag{5.3}$$

- The actor's k value is updated with the maximum of the k_u values.

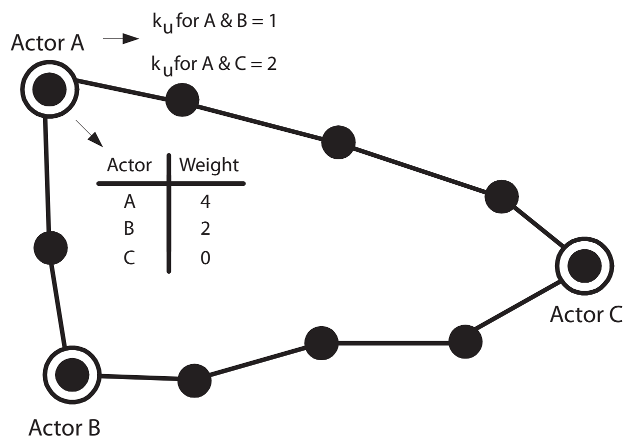


Figure 5.5: An example of finding k_u in a network with $k_n = 4$.

An example of deciding on k_u vaues in a network with $k_n = 4$ is given in Fig. 5.5. Actor A calculates the k_u values for actor B and actor C depending on the weights of these actors at actor A gathered by the network organization. These are the minimum k_u values kept in the records for each neighbor actor.

There are several aspects of k-adaptation which are critical for the algorithm. First, an actor's affiliation area is limited by adapting k, which reduces the number of packets that sensor nodes transmit as they update their weights since they have fewer actors to maintain. As the packet transmission drops, the energy consumption also decreases in the network. The other important aspect of k-adaptation is the hop-distance estimation since the values calculated with this method are used when estimating the locations of the sensor nodes. Moreover, many multi-hop localization algorithms in the literature become ineffective as the topology becomes irregular. Our algorithm's adaptation of k forces sensor nodes to be affiliated only with actors in close proximity which reduces the negative impacts of the irregular topology.

The packets originated from a sensor node v are transmitted to an actor a over a path of $h_a(v)$ hops. Hence the maximum number of transmissions in the actor area of a with n sensor nodes would be defined as $\sum_{i=1}^{n} h_a(v_i)$ when each sensor node transmits a packet. Then the improvement of the consumed energy in an actor area can be defined as follows:

$$E_i = \sum_{i=1}^{n} h_i - \sum_{j=k_a}^{n} h_j \tag{5.4}$$

The number of nodes in an actor area (n) decreases as the k value is adapted according to the neighbor actors. If the distance to the furthest neighbor actor is d_a, the adapted k value (k_a) can be assumed to be $\frac{d_a}{2}$. Then the number of sensor nodes affiliated with an actor can be approximated in proportion to the change in the actor area, as $n\frac{{d_a}^2}{4r^2k^2}$ improvement of the consumed energy in an actor area is denoted as follows:

$$E_i = h_{av}(n - n\frac{{d_a}^2}{4{r_s}^2k^2}) \tag{5.5}$$

where h_{av} is the average hop value of the sensor nodes, which are not affiliated with the actor a with the adaptation of k. In a uniform distribution of the sensor nodes, h_{av} is greater than $\frac{1}{2}(\frac{d}{2r_s} + k)$. Hence the improvement in energy can be approximated as follows:

$$
\begin{aligned}
E_i &= \frac{1}{2}\left(\frac{d_a}{2r_s} + k\right)\left(n - n\frac{{d_a}^2}{4{r_s}^2k^2}\right) \\
&= \frac{nd_a}{4r_s}\left(1 - (\frac{d_a}{2r_sk})^2 - \frac{d_a}{r_sk}\right) + \frac{nk}{2}
\end{aligned} \tag{5.6}
$$

The adaptation of k is also important for the hop-distance estimation since the values calculated with this method are used when estimating the locations of the sensor nodes. The adaptation of k forces sensor nodes to be affiliated only with actors in close proximity which reduces the negative impact of the dynamically changing topology.

An actor keeps different k values for all the other actors that they store weight values for. These values are used in the calculation of "1-hop distance". The average 1-hop-distance value is estimated as $\frac{d}{2k_u}$ and this calculated value is used for localization purposes. The sensor nodes dynamically store and update their weights as they float in the network. Actors collect weight information when the data packets are transmitted to the actors. Sensor nodes piggyback their weight tables to the data packets and these tables are used to estimate the distances of the nodes to the actors. The estimated distance of a sensor node to an actor A is calculated as follows:

$$d_A = (k_A - w_A) * h_A \tag{5.7}$$

where d_A is the estimated distance to the actor A, w_A is the weight of the sensor node for the actor A and h_A is the average 1-hop distance of actor A.

5.1.4 Mobility

The monitoring network aims to collect data from the unreachable parts of a river by allowing sensor nodes to drift with the force of the river currents. This uncontrolled motion of the sensor nodes follow the trajectories of the fluid parcels, making the mobility pattern more complex than the traditional random way point mobility or the group mobility models. The behavior of the network in the natural conditions of the river cannot be modeled with an assumption of a simple mobility pattern for the sensor nodes. Therefore, the movements of the sensor nodes must be modeled according to the properties of the river currents.

The motion of the subsurface currents in a river has several characteristic features. The advection in a river is affected from the variations in water depth, channel geometry, and the surface conditions. In a relatively short section of the river, the terrain can be assumed to force a higher volume of flow in the center line. Therefore, the velocity of the particles or the sensor nodes at the central stream will be higher than the ones closer to the boundaries. Furthermore, roughness of the surface may lead to eddies where the sensor nodes will be captured for a while and then possibly released to join the remaining part of the network.

To model these properties, we adopt the motion of the subsurface currents, initially employed in [99, 100].

The channel geometry of the river will mandate time varying meander amplitude which is denoted as follows:.

$$B(t) = B_0 + \epsilon \cos(\omega t + \theta) \tag{5.8}$$

Here, ϵ determines the degree of chaotic advection, where for relatively large ϵ, the particles are able to cross the jet in north to south direction or vice versa [163]. To model the currents in a river, we select ϵ such that the sensor nodes are allowed to mix with the jet stream. The direction of flow is denoted by ζ and it is defined as follows:

$$\zeta = \tan^{-1}\{Ak \cos[k(x - c_x t)]\} \tag{5.9}$$

Then, the motion of the sensor nodes can be defined with the following standard stream function [164]:

$$\Gamma(x, y, t) = \Gamma_0 \left\{ 1 - \tanh\left[\frac{y - \eta}{\xi / cos(\zeta)} \right] \right\} \tag{5.10}$$

where ξ is the width of the jet and $2\Gamma_0$ is the total eastward transport.

The time-varying central streamline is defined by η and it is given by the following equation:

$$\eta = B(t)\sin[k(x - c_x t)] \tag{5.11}$$

where k is the wave number defined in relation to the wavelength L, as $k = 2\pi/L$, and c_x is the phase speed of the sinusoidal meander [165].

where ξ is the width of the jet and $2\Gamma_0$ is the total eastward transport.

Eq. (5.10) represents the velocity field for an isopycnal surface where it is relevant to assume that sensor nodes can easily adapt to pressure changes since they already carry pressure sensor nodes as part of their monitoring task. To obtain the non-dimensional eastward moving frame, we substitute $x' = x - c_x t$ in ξ and in Eq. (5.11). Thus, Eq. (5.10) takes the following form:

$$\Gamma(x', y') = \Gamma_0 \left\{ 1 - \tanh\left[\frac{y' - \eta'}{\xi / cos(\zeta')} \right] \right\} + c_x y' \tag{5.12}$$

From Eq. (5.12), the velocity of a sensor node is computed by a simple derivation as follows:

$$u = \frac{dx}{dt} = -\frac{\partial \Gamma}{\partial y}, v = \frac{dy}{dt} = \frac{\partial \Gamma}{\partial x}. \tag{5.13}$$

In addition to the subsurface current mobility model, a basic directional mobility model is used to investigate the performance of the localization algorithm in different mobility conditions. In this second mobility model, each sensor node floats in the watercourse from a predetermined origin on one side of the river to a randomly chosen destination on the other side on a linear path.

5.2 Localization

The distance estimation is based on the number of hops needed to reach from the sensor nodes to the actor nodes. Therefore it's similar to the distance vector (DV) based approach of Niculescu and Nath's model [76]. However, Niculescu and Nath's model is designed for static networks and additionally, in most of the DV-based solutions, sensor nodes need to assign a fixed memory to save the locations of all the landmarks, hop-counts to these landmarks and average hop-distance values. In our approach, a sensor node keeps only the weight values for its affiliated actor nodes. All the other computation and memory requirements of the algorithm are handled by the actors and the sink, which is a better-fit for WSAN structure in terms of the usage of memory, computational resources and energy. Hereby, the information flooding, which is common and intense in DV-based solutions, is also minimized.

When the sensing sensor node transmits its weight values for three actors, its distances to these actors are estimated according to the "1-hop distance" value and the weights. Then, its position is calculated based on the acquired distances and the localization information obtained by the lateration.

The estimated distance of the sensor node is calculated for each of its affiliated actor and then these values are plugged into the lateration operation for the estimation of the sensor nodes position. When a sensor node is affiliated with multiple actor nodes, its position estimation can be represented with a system of equations, written in matrix form as follows:

$$\begin{bmatrix} 2(x_n - x_1) & 2(y_n - y_1) \\ \vdots & \vdots \\ 2(x_n - x_{n-1}) & 2(y_n - y_{n-1}) \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} (d_1^2 - d_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (d_{n-1}^2 - d_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix} \tag{5.14}$$

where (x_i),$y_i)$, $i = 1, ...n$, are the positions of the n actors and d_i, $i = 1, ...n$, are the estimated distance values.

The solution pair of this system, $(x_s$,$y_s)$,which minimizes $\|\boldsymbol{Ax} - \boldsymbol{B}\|_2$, is the pair that minimizes the mean square error, where $0.5\mathbf{A}$ is the left hand side matrix, $\mathbf{B}$ is the right hand side matrix and $\mathbf{x}$ is the vector for $(x_s$,$y_s)$ pair. Since $\|\boldsymbol{.}\|_2$ is minimized, the system is solved with minimum average error for all positions of the actors that the sensor is affiliated with.

For any vector $\boldsymbol{v}$, $\|\boldsymbol{v}\|_2^2$ is equal to $\boldsymbol{v}^T\boldsymbol{v}$, which can be used to find a solution for $\mathbf{x}$. Therefore, if the same expression is written for $\|\boldsymbol{Ax} - \boldsymbol{B}\|$, an equation for $\mathbf{x}$ can be found:

$$\|\boldsymbol{Ax} - \boldsymbol{B}\|_2^2 = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{Ax} - 2\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{B} \tag{5.15}$$

This expression is minimized when the mean square error is minimized. Therefore, the gradient of the expression has to be set to zero considering it as a function of $\mathbf{x}$:

$$2\boldsymbol{A}^T\boldsymbol{Ax} - 2\boldsymbol{A}^T\boldsymbol{B} = 0 \Leftrightarrow \boldsymbol{A}^T\boldsymbol{Ax} = \boldsymbol{A}^T\boldsymbol{B} \tag{5.16}$$

This is the normal equation for the linear least squares problem and it has a unique solution. We use the Cholesky factorization to solve this equation and to get the estimations for x and y coordinates of the sensor nodes. Accordingly, the resulting matrix is given by:

$$\boldsymbol{x} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{B} \tag{5.17}$$

The estimations for the coordinates of the sensor nodes are calculated by this method for each time instance and recorded. Since our goal is not an online location-awareness for the sensor nodes, the path the nodes followed is estimated after all of these points are calculated. Essentially, the paths are estimated using the interpolation of these coordinates. Hence, the effect of the errors in the individual estimates to the paths of the sensor node is minimized.

5.3 Simulation study

5.3.1 Simulation environment

We evaluate our approach by measuring how the estimated location errors vary with different network and algorithm characteristics. The number of the nodes and the value of k are used as the simulation parameters in the experimental setup. The actor nodes are stationary, deployed in the area uniformly at random with the constraint that they are able to form a connected graph for communication among themselves. Sensor nodes are mobile and they are flowing from left to right in the watercourse. The communication links among the sensor nodes are open to failure. Table 6.4 summarizes the simulation parameters used in our simulation study.

Table 5.1: Simulation parameters

Parameter	**Value**
Number of sensor nodes	1–25
Number of actor nodes	25–50
k value	1–5
Total area	250×250 meters
Sensor node transmission range	40 meters
Sensor node floating speed	1.25–1.5 m/s
Traffic type	Constant rate
Data packet size	256 bytes
Transmit power (W)	5.13×10^{-6}
Reception power threshold (dBm)	-95
Channel settings	Auto assigned
Short retry limit	7
Long retry limit	4

5.3.2 Simulation results

5.3.2.1 Experiment 1

In order to investigate the effect of clustering on the performance of the proposed algorithm, a scenario with only a single sensor node and 25 actor nodes is considered. The main purpose of this experiment is to observe the performance difference between 1-hop clustering and multi-hop clustering. In this scenario, a single sensor node floods from left to right with a speed between 1.25 to 1.5 m/s and connects to the actor nodes whenever it is in the transmission range. Hence the sensor node can be connected to multiple actors and the parameter k has no influence on the location estimation in this experiment.

Figures 5.6-5.19 show the exact trajectory of the node on the x and y coordinates respectively. The x-axis of the figures represents the time steps. The node is moving from 0 to 250m in x coordinate and for y coordinate the node stays on the position around 110m. The network density is sufficiently high enough that a position could be calculated for each time step. In other words, the node was affiliated with four actors for most of the time. In Figures 5.6-5.19, the estimated trajectory has been produced using linear regression over the data set of the estimated positions. The estimated positions are with an error of 30 meters according to the exact trajectory, but most estimated positions show much less error. The error is as expected since the transmission range is 40 meters and the estimation is based on the estimated hop-distance. However, the estimated trajectory of the considered sensor nodes is within a range of at most 10 meters, but mainly within 5 meters or less.

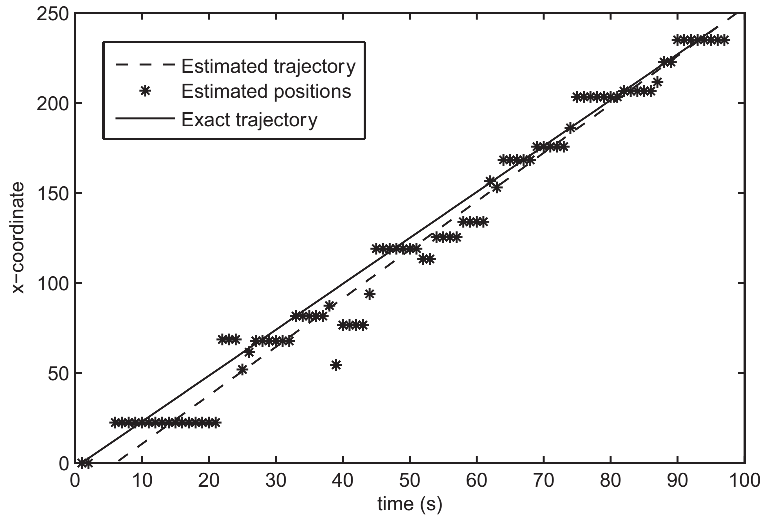


Figure 5.6: Estimated and real x-coordinates for 1 sensor node and k=1.

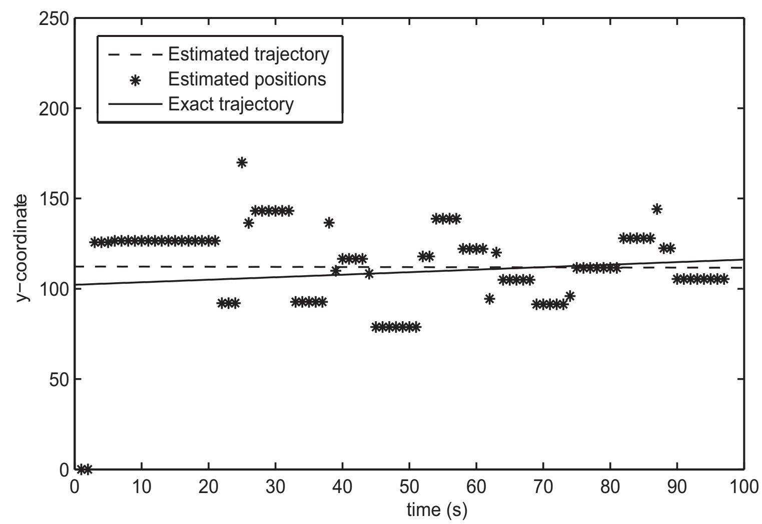


Figure 5.7: Estimated and real y-coordinates for 1 sensor node and k=1.

The settings in the 1-hop clustering experiment have been modified such that 25 actors and 25 sensor nodes are participating in the simulation area and the parameter k is set to three. In this case, the maximum length from a sensor node to an actor node can be three hops. In similar illustrations as in Figures 5.6-5.19, the accuracy of the location estimation of one of these 25 nodes under the use of local k-hop clustering information is shown in Figures 5.8-5.9. The sensor node is connected to the actor networks via intermediary nodes, where k=3, for most of the time in this experiment. Therefore multi-hop information is used. The applied algorithm estimated the hop-distances using far less accurate data. However the results show that the estimated positions are very close to the exact trajectory. Compared to the previous experiment, the error is only slightly higher. The main difference is that there are more time steps where there is not enough information to estimate the position. It should be noted that this result also depends on the real trajectory of the individual sensor node.

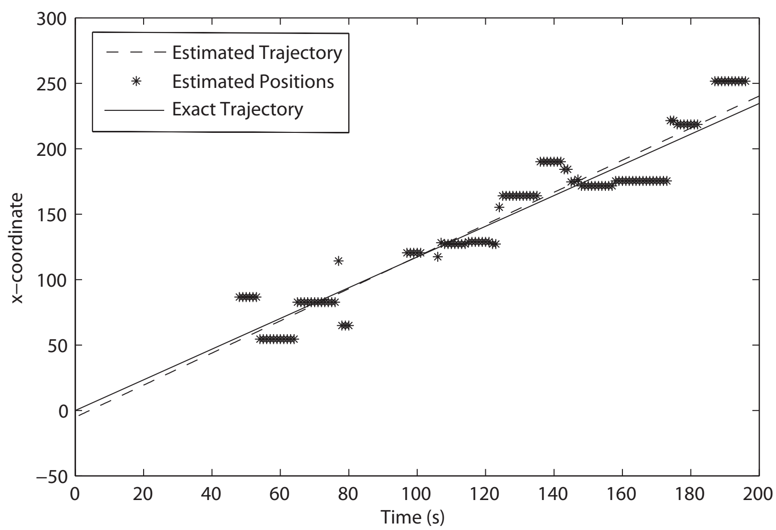


Figure 5.8: Estimated and real x-coordinates for the experiment with k=3.

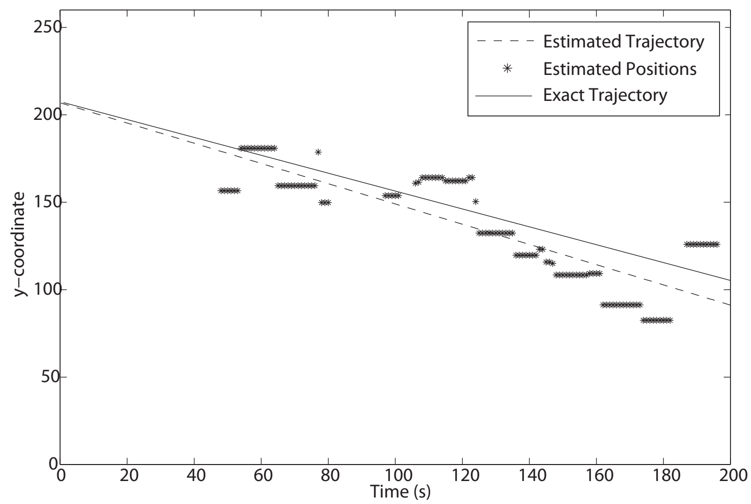


Figure 5.9: Estimated and real y-coordinates for the experiment with k=3.

5.3.2.2 Experiment 2

In this setting, the number of nodes is 50, 25 of which are the actors and 25 are the sensor nodes. However, in order to get the accuracy for single hop settings, the parameter k is set to 1. The error distribution is illustrated in Fig. 5.10. The error has a normal distribution and it ranges from -30 to 30 meters. Hence this error is for the x-coordinate estimation. Since the nodes are floating almost horizontally, the accuracy on the y-coordinate is appropriate. We can also observe that the number of estimated points is not as high as in the previous experiment.

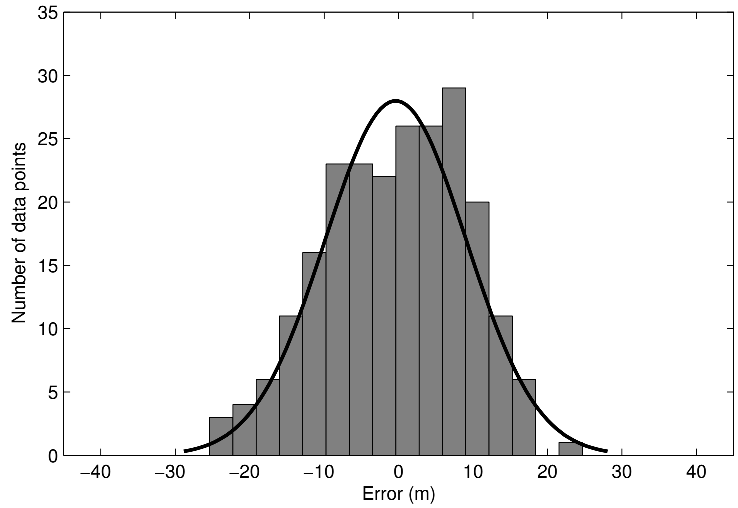


Figure 5.10: The error distribution of the results for the experiment 3 with k=1.

5.3.2.3 Experiment 3

The multi-hop clustering approach for location estimation is a core part of the contribution of the proposed localization algorithm. In this experiment, in order to investigate the usage of multi-hop clustering, the number of sensor nodes is increased while the number of actor nodes remains equal. Thus actors, which are not in the transmission range of a single mutual sensor node, must make use of the multi-hop information provided by the passing sensor nodes. The k parameter is set to three allowing the nodes to connect to an actor through another sensor node. The resulting Fig. 5.11 shows that the error distribution is similar to the Fig. 5.10 where the sensor nodes were connected directly to the actor nodes. The error rate is again normal distribution. However, there are two differences: (i) more positions could be estimated which means that the local multi-hop approach applied in our algorithm is able to gather more localization estimation than it does with a traditional approach, and (ii) the estimation accuracy is decreasing. Compared to the experiment 2, where the distribution was in between -30 and 30 meters, the distribution is now in between -60 and 60 meters.

We observe that higher the value of k is, more positions can be estimated by our algorithm. However, the accuracy decreases at the same time with the increasing k. Therefore the selection of k is an important issue for the network, which depends on the number of nodes, the environmental conditions and the application requirements.

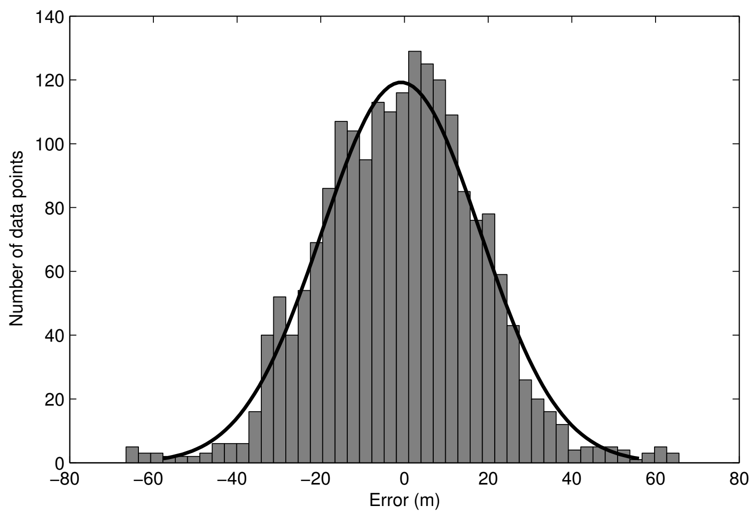


Figure 5.11: The error distribution of the results for the experiment 4 with k=3.

5.3.2.4 Experiment 4

In this experiment, the settings remained the same as the experiment 3 except for the value of k. The value of k is increased to five, so clusters with more hops compared to the experiment 3 are allowed in the network. The error distribution of the experiment is given in Fig. 5.12.

Similarly, Fig. 5.12 shows that as the value of k becomes higher, the accuracy decreases further while the number of estimations increases due to further connection chains in the network.

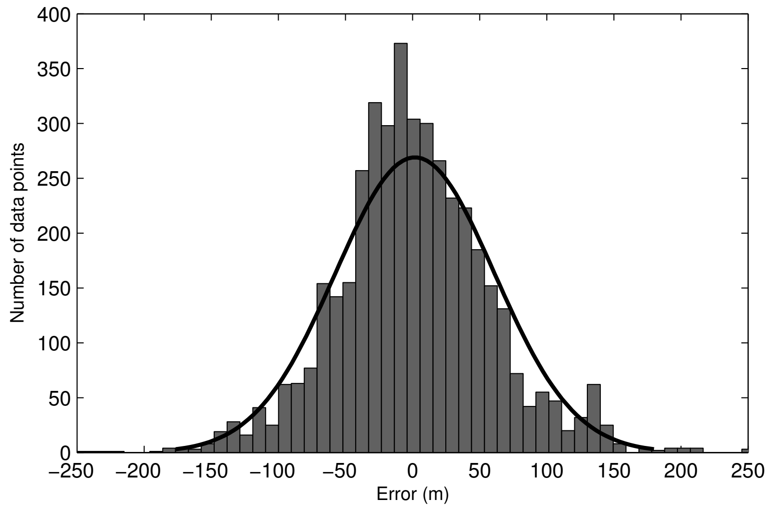


Figure 5.12: The error distribution of the results for the experiment 5 with k=4.

5.3.2.5 Experiment 5

In the following experiments, sensor nodes are modeled to be flowing in the watercourse with the basic directional mobility model and the meandering mobility model. The communication links among the sensor nodes are assumed to be prone to failures due to shadowing and multipath caused by the river environment.

In the first set of experiments, the parameter k is set to a small value ($k = 2$) to observe the performance of the algorithm in a low clustering scenario. This value minimizes the influence of clustering by limiting the maximum number of hops between a sensor node and

Table 5.2: Simulation parameters

Number of sensor nodes	100
Number of actor nodes	25
k value	1-5
Terrain size	200×1400 m
Sensor node transmission range	40 m
Average jet speed	1-7 m/s
Degree of chaotic advection	0.3
Frequency of time dependent meanders	0.4
Simulation time	3600 s

its affiliated actor node. Therefore, in this case, only the sensor nodes, which can reach the actor through one or two hops are allowed for affiliation. The experiments are conducted for both the meandering and the basic directional mobility models. The error distribution of localization with the basic directional mobility (BDM) is illustrated in Fig. 5.13. The error distribution when using meandering mobility (MM) model with average jet speed of 1.5 m/s is shown in Fig. 5.14.

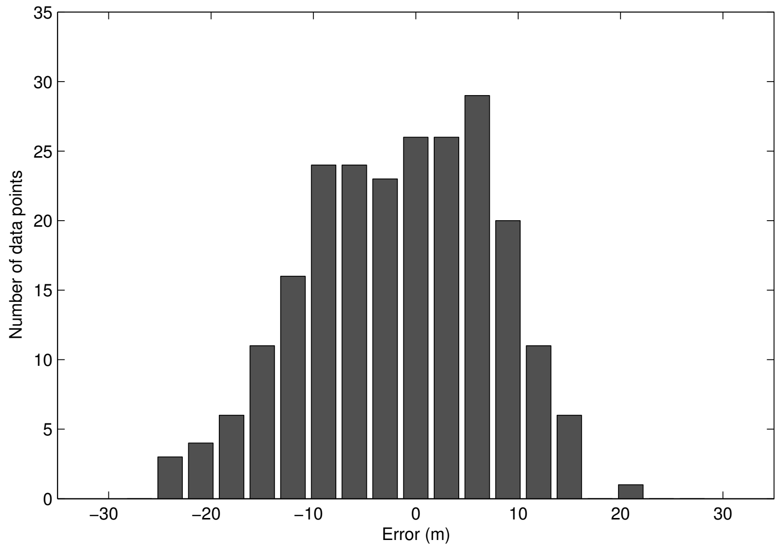


Figure 5.13: The error distribution of the experiment 6 with BDM and $k = 2$.

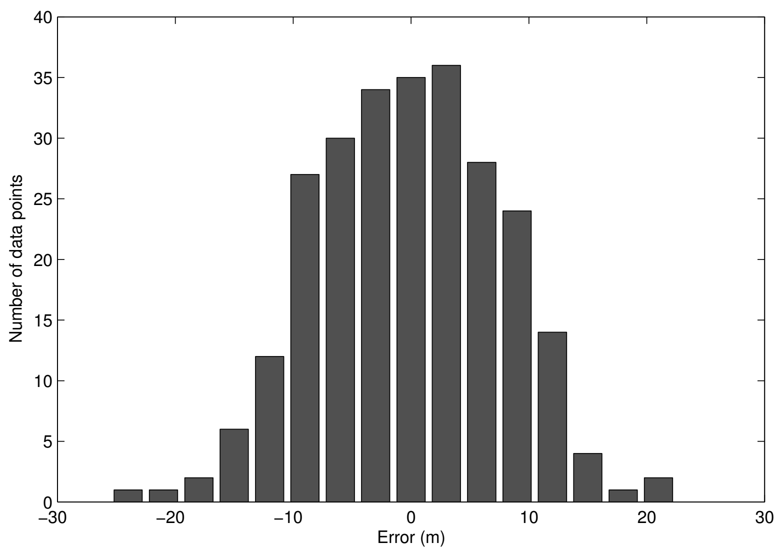


Figure 5.14: The error distribution of the experiment 6 with MM and $k = 2$.

In the second set of the experiments, the value of the parameter k is increased to five. The goal of these experiments is to observe the influence of clustering in the location estimation by comparing the results of the first set to the second set of experiments. The location estimation error distributions are illustrated in Fig. 5.15 and Fig. 5.16 for basic directional and meandering mobility models, respectively.

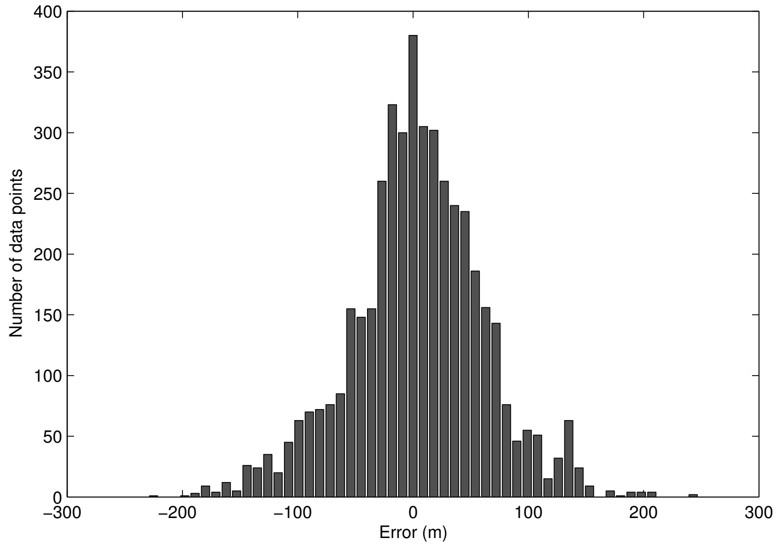


Figure 5.15: The error distribution of the experiment 6 with BDM and $k = 5$.

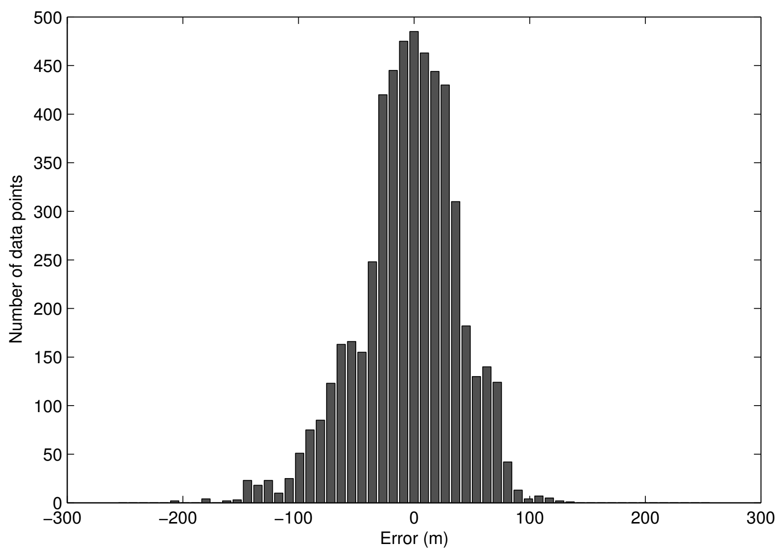


Figure 5.16: The error distribution of the experiment 6 with MM and $k = 5$.

The localization error has a normal distribution for both values of the parameter k. When $k = 2$, the range of the error is between -25 and 25 meters for both mobility models. The estimation errors are equal to or below 15 meters for $\sim 93\%$ of the results in basic directional mobility and $\sim 97\%$ of the results in meandering mobility model. When k is increased to five, the range of the error results is more than the results observed in the first experiment. Since the number of affiliations increases as k increases, we also observe that the number of estimated points is higher when k is five. When the meandering mobility model is used, the error is below 40 meters for 73% of the results. These simulation results show that as the length of communication paths are allowed to become larger, more positions can be estimated with a cost on the accuracy. Therefore, the selection of the maximum hop number depends on the requirements of the network under consideration and the scenario.

5.3.2.6 Experiment 6

In this experiment set, the impact of the mobility model on the performance of the localization algorithm is evaluated. The movement of sensor nodes is modeled with meandering and the basic mobility models, where the parameter k is set to three and the average jet speed used is 1.5 m/s. The error distribution results are illustrated in Fig. 5.17 and Fig. 5.18 for basic directional and meandering mobility models, respectively.

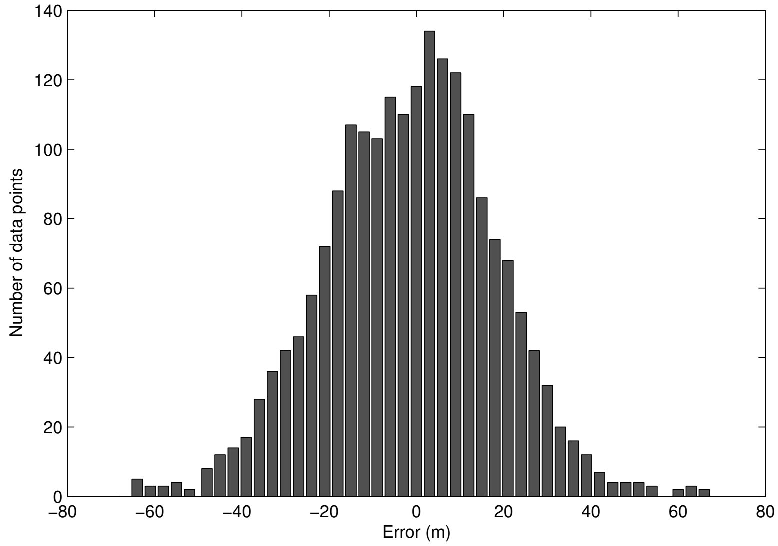


Figure 5.17: The error distribution of the experiment 7 with BDM and $k = 3$.

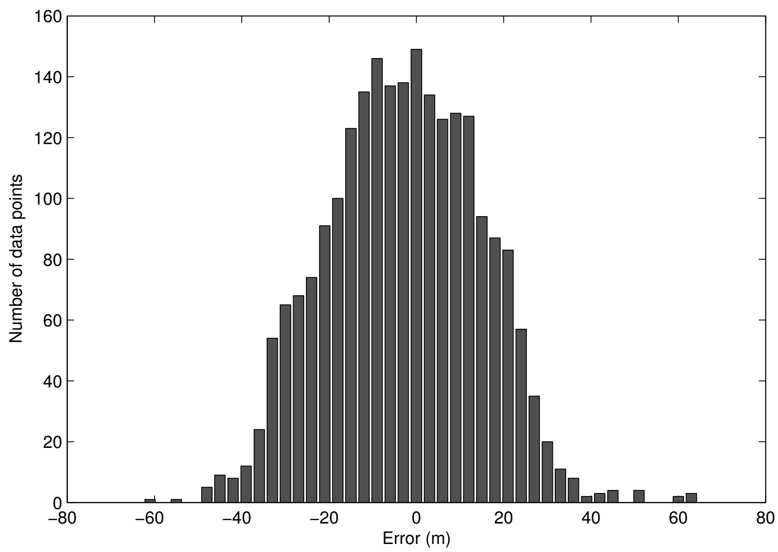


Figure 5.18: The error distribution of the experiment 7 with MM and $k = 3$.

Using the results of this experiment set and the previous one, from Fig. 5.13 to Fig. 5.18, we observe that the error has a normal distribution for both models, while the number of data points is larger for meandering mobility model. The difference in the number of data points is higher for low error values in favor of the experiments with meandering mobility model, which shows that the accuracy of the algorithm is better with meandering mobility model.

The location estimation is also demonstrated for particular sensor nodes in the experiments to observe the accuracy of the algorithm. Fig. 5.19 and Fig. 5.20 show the exact trajectory of one of the sensor nodes on the x coordinate and the estimations of our algorithm for both mobility models. Fig. 5.21 and Fig. 5.29 show the results of the experiment for y coordinate.

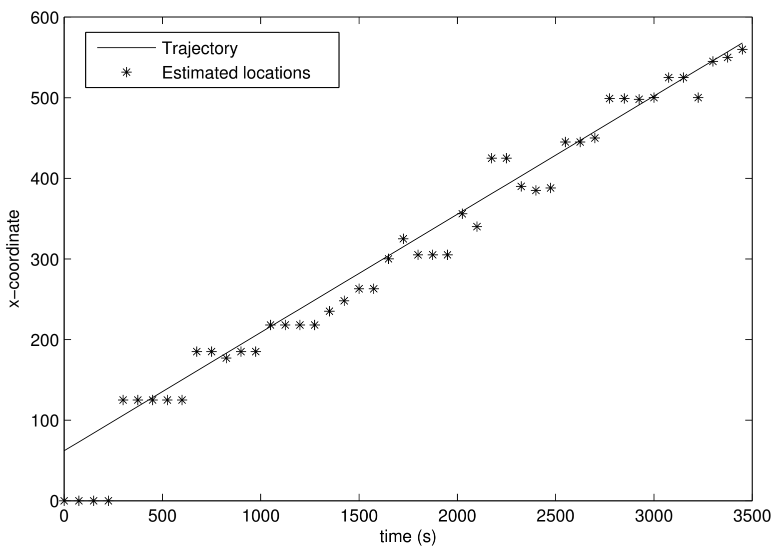


Figure 5.19: Estimated and real x-coordinates of a sensor node with BMM.

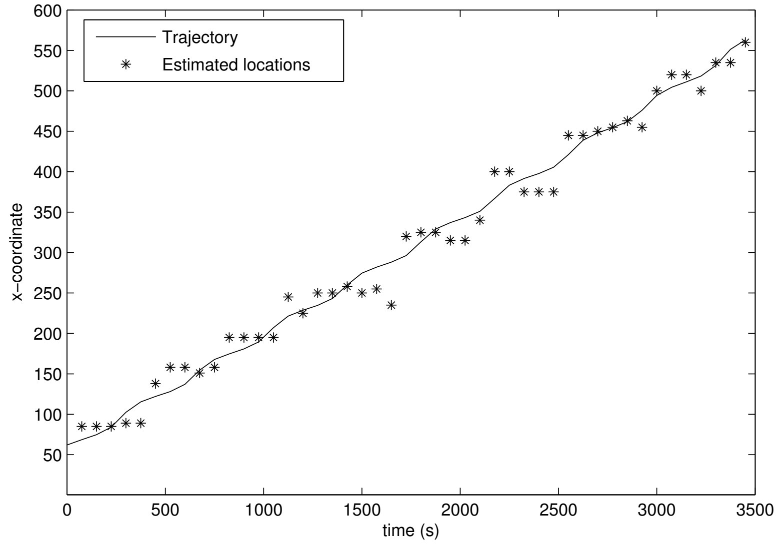


Figure 5.20: Estimated and real x-coordinates of a sensor node with MM.

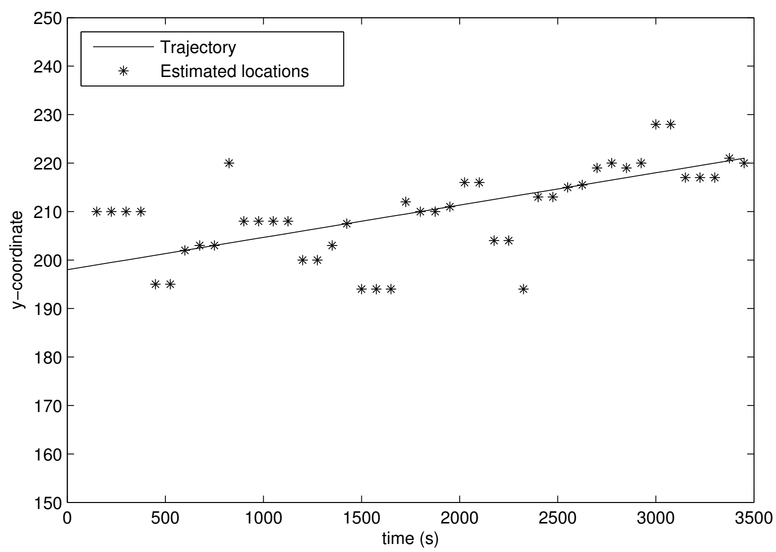


Figure 5.21: Estimated and real y-coordinates of a sensor node with BMM.

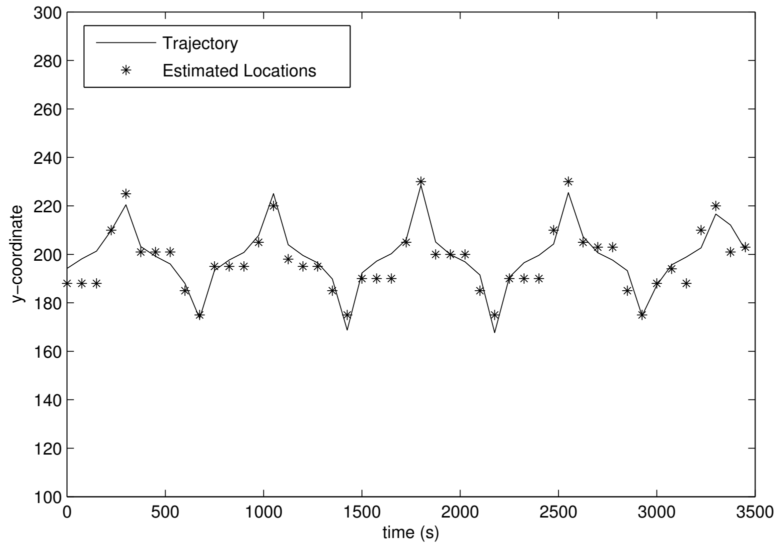


Figure 5.22: Estimated and real y-coordinates of a sensor node with MM.

As it was observed with the error distribution results, the accuracy of the estimation results is higher with meandering mobility model. This is because of a stretching topology with correlated motion where some sensor nodes float behind others supplying reach for more actors when compared to random floating of the nodes in basic directional mobility. The localization algorithm gives better results when the sensor nodes move with meandering mobility. This is an advantage of the localization algorithm and denotes that the algorithm would have better performance in a real-life scenario compared to a random mobility simulation due to realistic characteristics of the meandering mobility model.

The cost of our localization approach in terms of energy consumption is demonstrated in Fig. 5.23 with 95% confidence interval for different values of k. The energy consumption is observed for basic directional mobility model (BDM) and for meandering mobility model with (MMk) and without (MM) k adaptation. The energy consumption parameters of this simulation set are chosen from the multi-sensor system, which was deployed in the River Lee Co. Ireland [9]. Similar to our application scenario, this multi-sensor system is used to monitor water quality parameters such as pH, temperature or dissolved oxygen. According to the power consumption analysis of Regan et al. [9], the power consumption of a node in the simulation is 96.2 mW when the transceiver is active and 0.054 mW when the transceiver is not active.

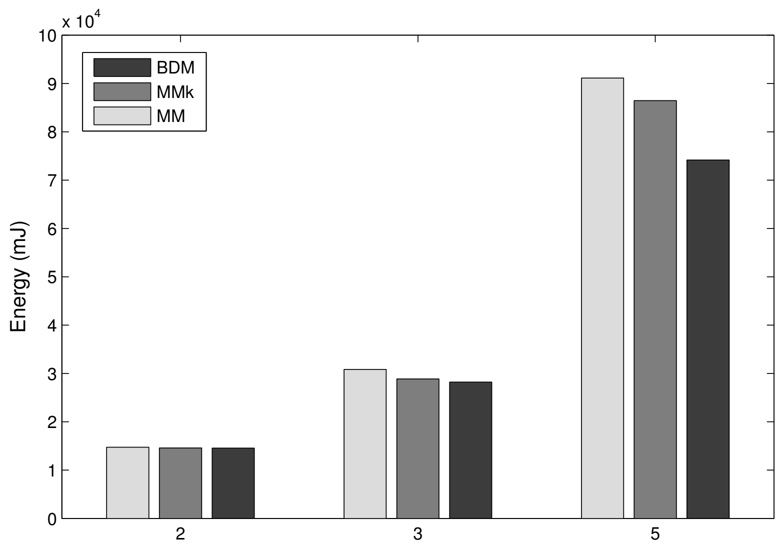


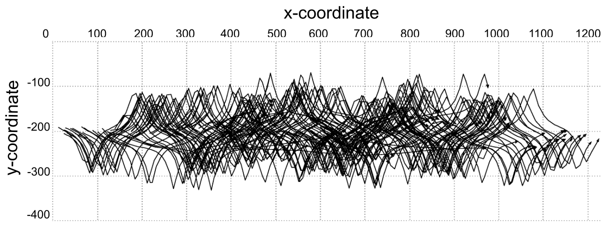
Figure 5.23: Energy consumption for the proposed models with different k values.

One of the important results denoted in Fig. 5.23 is the effect of k in the energy consumption. The energy consumption increases with the increasing values of k. As shown in the results of error distribution experiments, the collected data also increase with larger k values. Therefore, the number of data receptions and transmissions is larger for larger k values with the cost of an increase in energy consumption. Additionally, the energy consumption is higher for meandering mobility model and the difference accumulates as the value of k

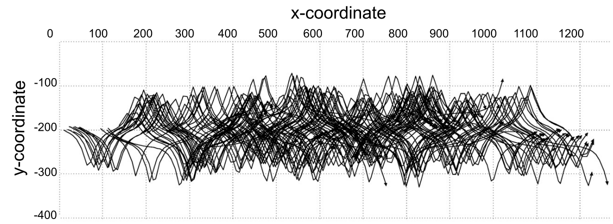
increases. Hence the higher clustered structure created by the meandering mobility model results in more data transmission and more energy consumption. The improvement in energy consumption by the adaptation of k is also observed in Fig. 5.23 and this effect becomes more apparent as k increases.

5.3.2.7 Experiment 7

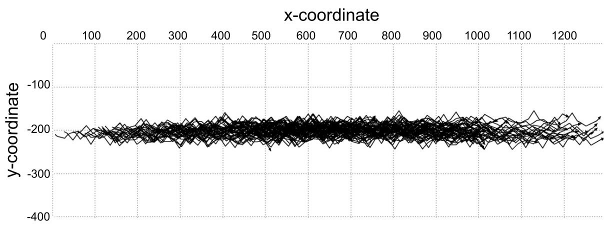
In the meandering mobility model, the average jet speed is an important factor to define the characteristics of the meandering behavior of the river current. For a comprehensive evaluation of the proposed localization approach, the average jet speed parameter of the meandering mobility model is varied in the next set of simulations.



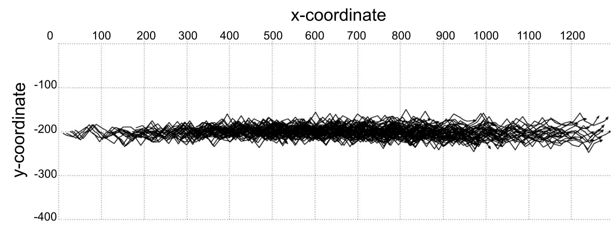
(a) Average jet speed = 1 m/s



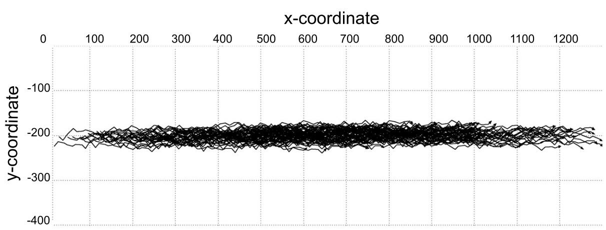
(b) Average jet speed = 1 m/s



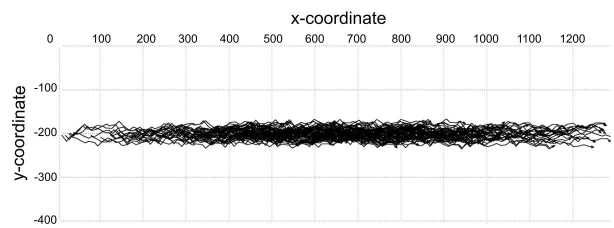
(c) Average jet speed = 3 m/s



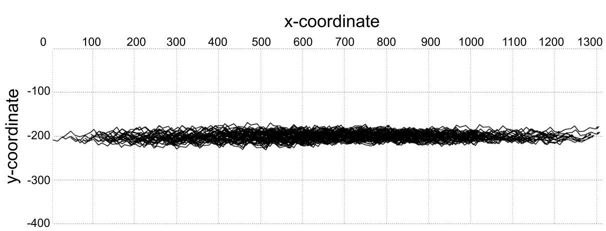
(d) Average jet speed = 3 m/s



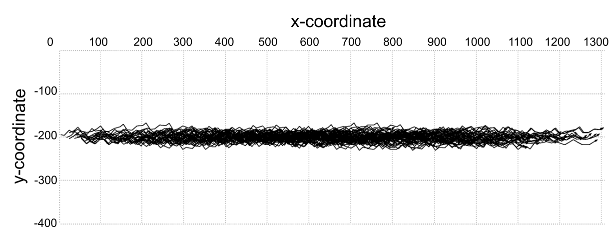
(e) Average jet speed = 5 m/s



(f) Average jet speed = 5 m/s



(g) Average jet speed = 7 m/s



(h) Average jet speed = 7 m/s

Figure 5.24: Examples for the paths of 100 nodes with varying average jet speed values.

Fig. 5.24 shows the paths for all sensor nodes in two examples from each of the simulation sets with average jet speed values varying from one to seven meters per second. Fig. 5.24 (a) and Fig. 5.24 (b) show the paths of nodes with the average jet speed of one meter per second. When the average jet speed is increased to three meters per second as presented in

Fig. 5.24 (c) and Fig. 5.24 (d), the effect of this increase can be observed particularly in the meander amplitude. The jet speed is further increased to five meters per second in Fig. 5.24 (e) and Fig. 5.24 (f) and seven meters per second in Fig. 5.24 (g) and Fig. 5.24 (h). The variance in the y-coordinates of the sensor node positions decreases as the jet speed is increased. This characteristic of the mobility model can be used to adjust the values of the localization algorithm parameters for the specific requirements of different application scenarios.

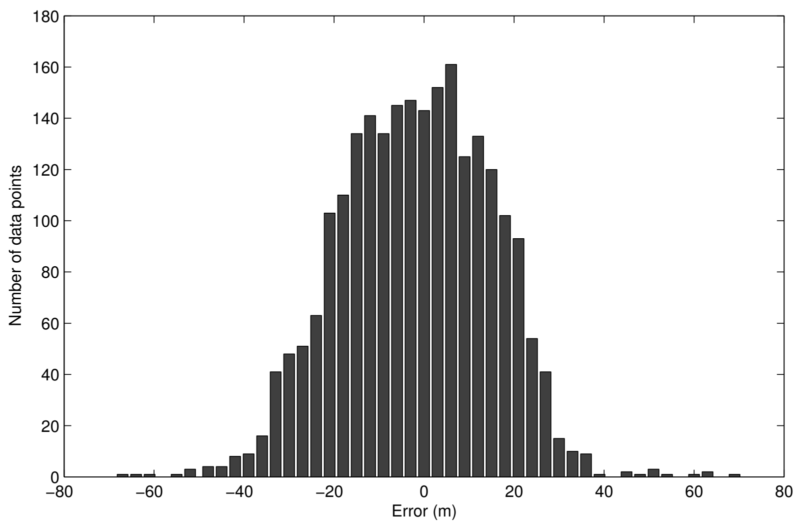


Figure 5.25: The error distribution with average jet speed = 1 m/s and k = 3.

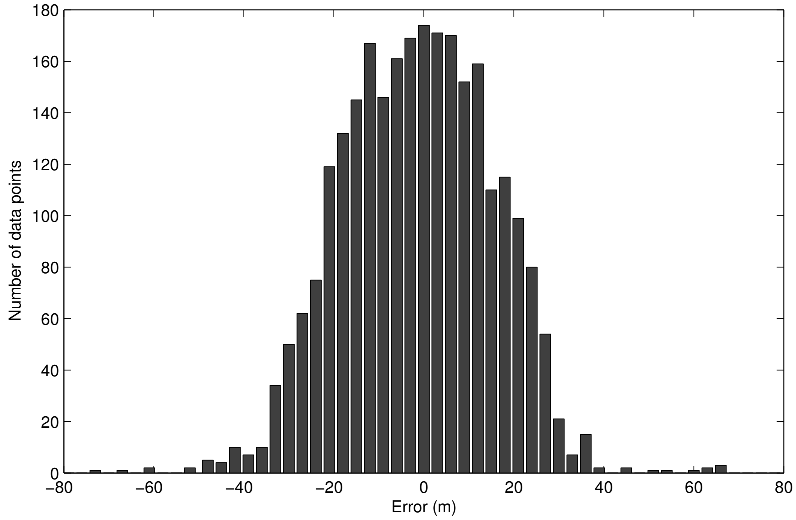


Figure 5.26: The error distribution with average jet speed = 3 m/s and $k = 3$.

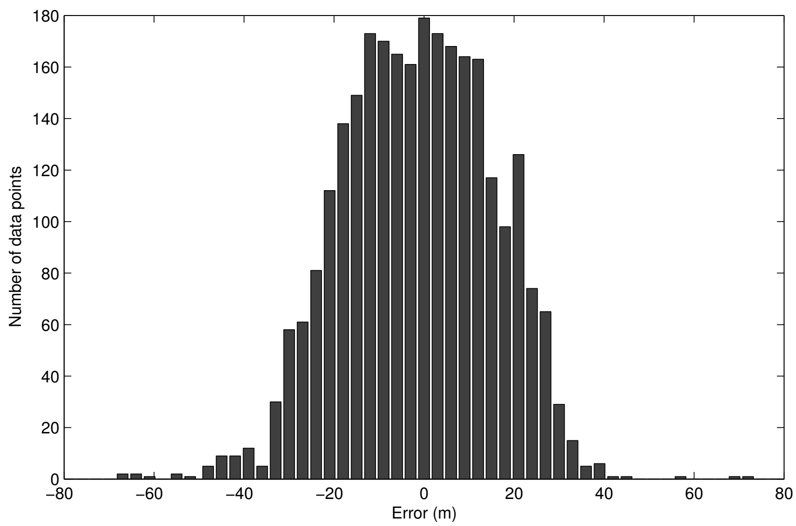


Figure 5.27: The error distribution with average jet speed = 5 m/s and $k = 3$.

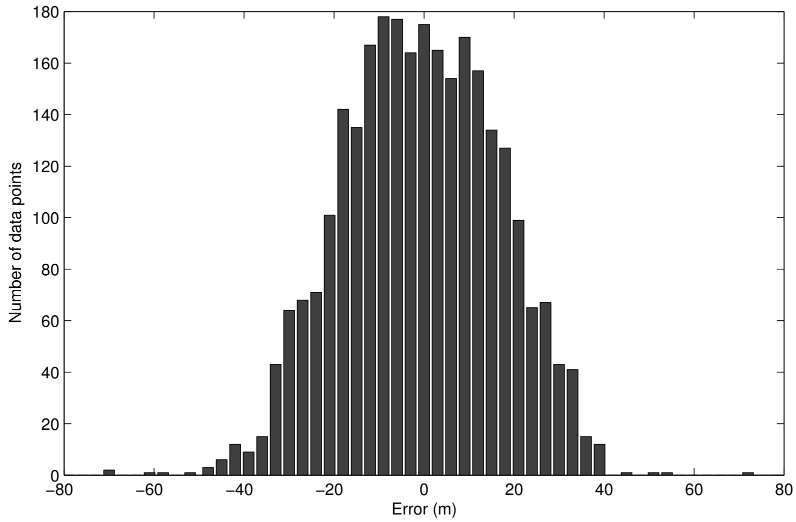


Figure 5.28: The error distribution with average jet speed = 7 m/s and $k = 3$.

Figs. 5.25-5.28 show the error distribution results of the location estimation for meandering mobility with the average jet speed values 1m/s, 3m/s, 5m/s and 7m/s, respectively.

The localization errors collected for the experiments with different average jet speed values have results close to each other. However the results also show that the accuracy of the position estimation increases for these four values with the increasing jet speed. The number of data points collected also increases as the meander amplitude decreases. This result can be explained with the additional stretching in the topology when the meandering jet amplitude decreases.

The effect of the variation in the average jet speed parameter of the meandering mobility model is also tested for the energy consumption. Fig. 5.23 shows the average energy consumption with 95% confidence interval for different values of average jet speed. When these results are combined with the results in Fig. 5.23, we observe that the energy consumption increases with the increasing values of both k and jet speed, and the selection of k value is more influential in terms of energy when utilizing the localization approach.

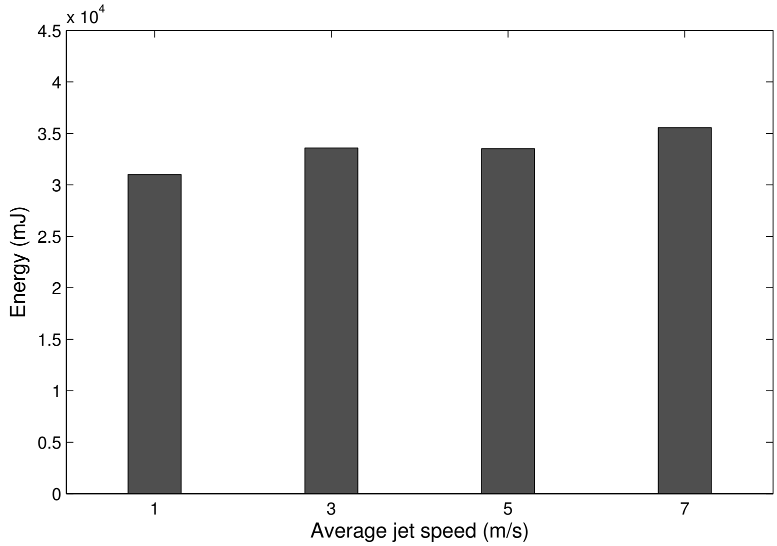


Figure 5.29: Energy consumption for MM with different average jet speed values.

5.3.2.8 Experiment 8

The settings chosen in the above set of experiments are to match the requirements of the Amazon scenario. Common settings, parameters and mobility models have been used to guarantee the reproducibility of the results. This configuration is a subject of refinement in future work. The Amazon scenario also puts restrictions in terms of node speed, node direction, connection behavior and location of actor nodes on the simulation model. In the next step, we strive to find out how far the introduced approach can be applied to other monitoring scenarios such as animal monitoring.

The scenario considered in the next set of experiments is an ape habitat. We assume that each ape is equipped with a wireless sensor such as the ones used in the Amazon scenario. However, a certain fraction of older and stronger apes are equipped with actors. On some rare points in the tribe habitat, gateway nodes are installed to collect data from the actor network when they are in the transmission range. In this new setting, the actors are mobile and the directions of all of the nodes are subject to change arbitrarily within a given area. We assume that the ape mobility pattern is described by the random waypoint model. More realistic mobility patterns exist in the literature, but in this chapter we assume the affiliations are interrupted permanently and we focus on the proposed localization approach.

An appropriate choice of the parameter k was important in the Amazon scenario. The results show that higher the parameter k, more flowing nodes are affiliated to the clusters and these nodes deliver more data. In the ape scenario, the simulation settings have been changed in a way that the parameter k always remains the same but the fraction of actors compared to the total number of nodes changes. Table 5.3 summarizes the results for successful estimations for different experiments in our simulation study, where the transmission range of a node is taken as 50 m and the distribution area is 300 m^2.

Table 5.3: Fraction of successful position estimation with respect to network density.

Total nodes	Number of actors	Fraction of actors	Successful estimations
50	10	1/5	1372/10000
50	20	2/5	2821/10000
50	30	3/5	2882/10000
100	10	1/10	2687/10000
100	20	2/10	6015/10000
100	30	3/10	6249/10000

The results in the Amazon scenario indicate that the number of actors is critical for the success of the estimation. However, Table 5.3 shows that a network configuration with even a small ratio of actors is able to deliver an appropriate number of estimation results. We understand this effect due to the increase of the network density implied by the increase of the number of nodes. Comparing the results for the experiments "50/10" and "100/20" for which the fraction of actor nodes is the same, we observe that the number of successful position estimation increases drastically while k remains the same. Therefore the quality of the estimation increases by increasing network density while keeping the number of actor nodes constant. The positive effect of increase in the fraction of actor nodes saturates at one point and do not result in a further increase in estimations. The influences of the network density, fraction of actor nodes and k can be observed in Table 5.3. The parameter k and the network density are identified as the factors behind the increased number of successful estimations. The network density can be increased by either increasing the number of nodes (while keeping the number actor nodes the same) or by reducing the distribution area. Two example trajectories and their estimations for a node are shown in Figures 5.30-5.31 and Figures 5.32-5.33.

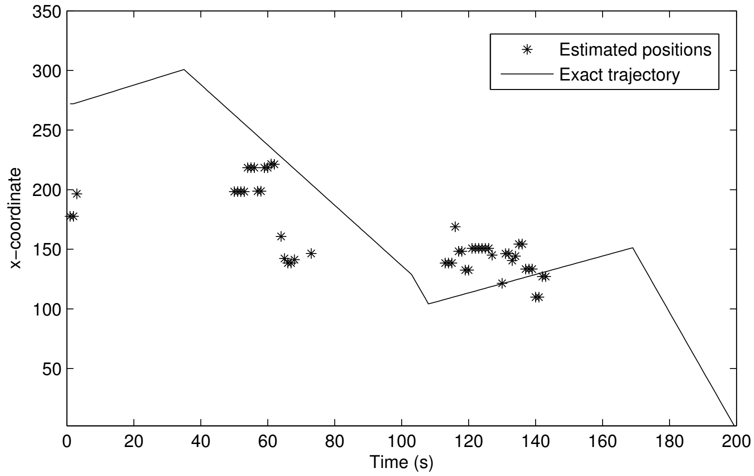


Figure 5.30: Estimated and real x-coordinates for experiment 8 "50/10" case.

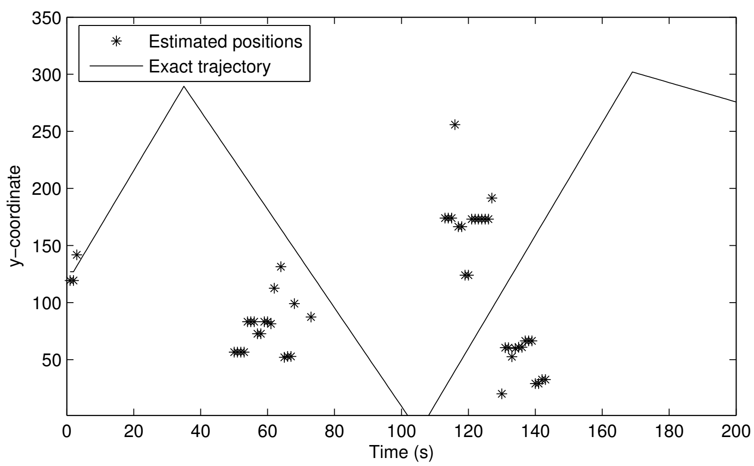


Figure 5.31: Estimated and real y-coordinates for experiment 8 "50/10" case.

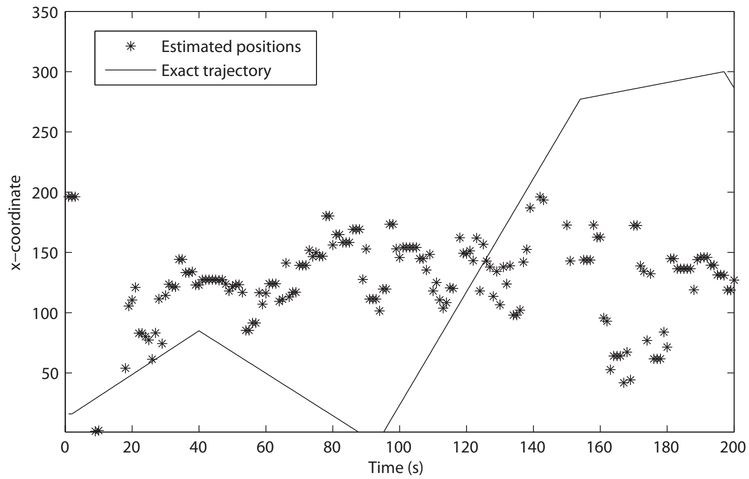


Figure 5.32: Estimated and real x-coordinates for experiment 8 “100/30” case.

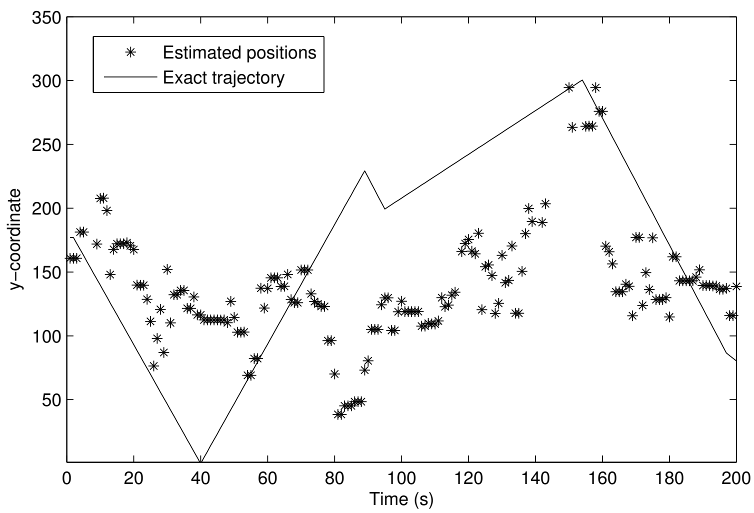


Figure 5.33: Estimated and real y-coordinates for experiment 8 “100/30” case.

It can be observed that the estimation position for sparse networks (or networks with low k) is often not possible, so the example with 50 sensor nodes shows only few successful estimations, although 40% of all nodes are actors. By increasing the number of nodes, i.e.

by increasing the network density, the number of successful estimations increase in spite of the fact that the fraction of actor nodes has been reduced.

5.3.2.9 Experiment 9

In this experiment, the same setting as in the previous experiment is used and the effect of the mobility of actors on the estimation accuracy is investigated.

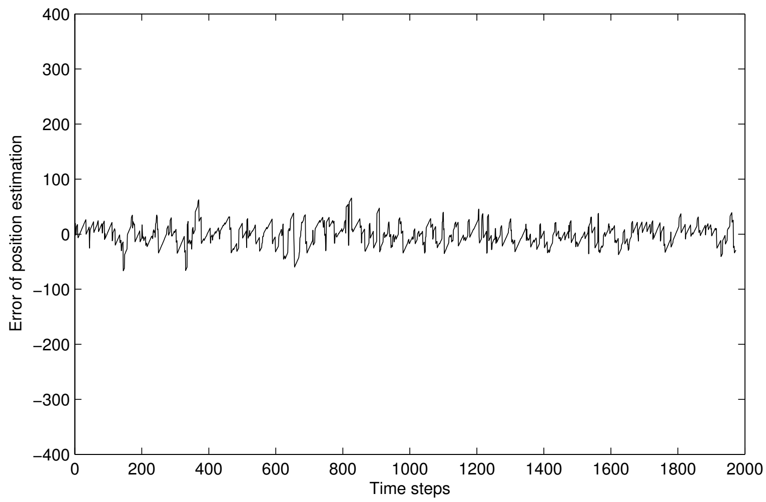


Figure 5.34: 30 sensor nodes with fluid mobility model and 20 static actors.

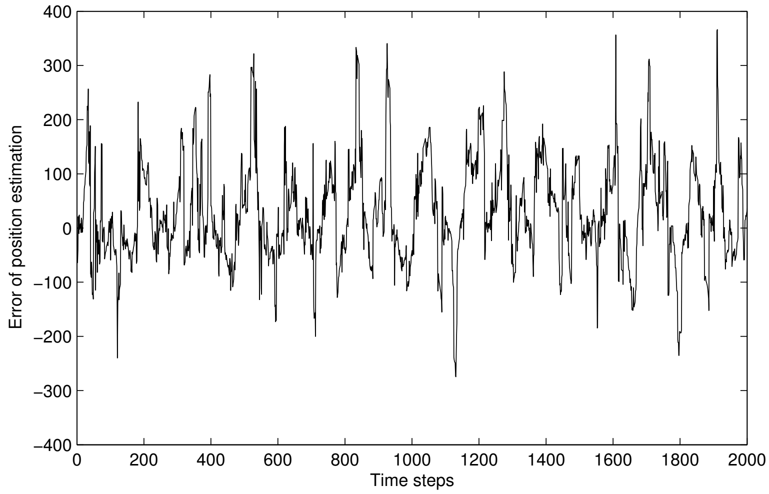


Figure 5.35: 30 sensor nodes and 20 actors with random waypoint mobility model.

There are two cases with 30 sensor nodes, 20 actors, $k = 4$ and deployed into the same simulation area of 300m^2. In the first case, the fluid mobility mode is used with static actors and in the second case the random waypoint model is applied for both sensor nodes and actors. It can be observed in Figures 5.34-5.35 that most of the estimations are in between -50 and 50 meters for both cases where the mobile setting gives a little less efficient results. The accuracy of the estimation is more precise in the static setting while the estimation error can be too large, making it unusable in the mobile setting. The more drastic changes observed in the accuracy on neighboring positions is the result of the influence of mobile actors on the estimation process. The actor mobility causes sudden changes in the information available to estimate a nodes position, which affects the accuracy.

5.3.2.10 Experiment 10

Although the accuracy changes drastically due to the mobile actors, the error behavior of the estimation remains as a Gaussian distribution and similar to the error distribution characteristics of the static experiment discussed above. Figures 5.36-5.37 show the error distributions in position estimations for 100 mobile nodes with 30 actors.

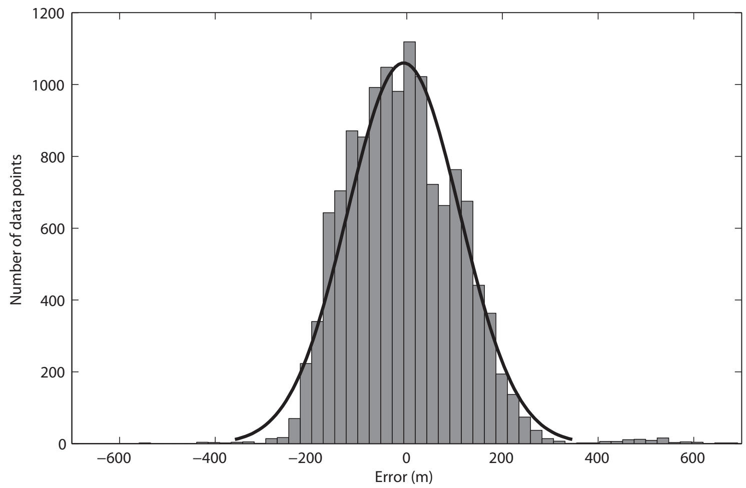


Figure 5.36: Error distribution in x-coordinates for “130/30” case.

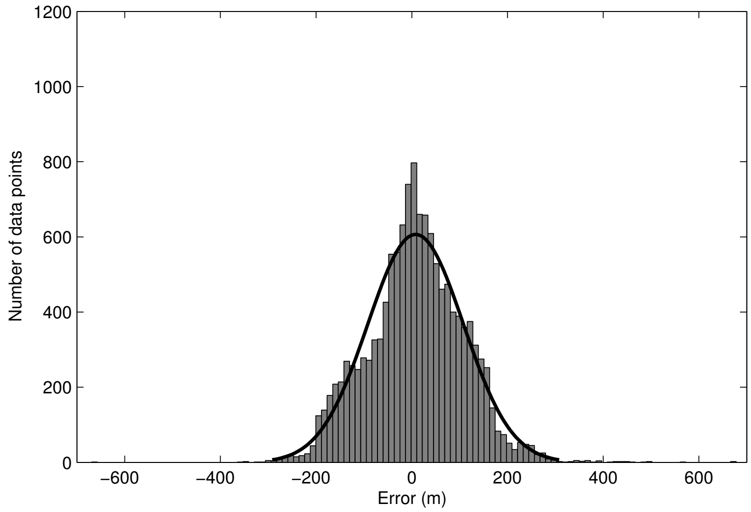


Figure 5.37: Error distribution in y-coordinates for “130/30” case.

Although the mobile actors have negative influence on the estimation accuracy, the drastic changes in the estimation accuracy also provide changes for improvement. When a node is affiliated with only three actors at time t, the estimation cannot be very accurate. The same nodes position might have been estimated more accurately (i.e. more than 3 actors could contribute information) at time $t-1$. Due to the drastic changes in the position estimation, it is feasible to detect less accurate estimation and correct them by using interpolation considering the neighboring points. By this way, the mobile actors can contribute towards making more accurate estimations.

CHAPTER 6
ANIMAL MONITORING

In this chapter, we present data collection and evaluation algorithms to provide a complete model of primate monitoring by using a WSAN. More specifically, we consider the modeling and monitoring of the social life of primate groups [19–21]. We study two main problems. First, we develop an algorithm for the determination of social roles in an animal society. Second, we approach the problem of the absence of realistic data to model the movement patterns from a social network perspective.

Animal monitoring becomes even more challenging when the observed species group possess a complex social structure as it requires simultaneous monitoring and assessment of multiple individuals and their interactions. For instance, the complex social organization of primates requires continuous and long-term monitoring to gather sufficient data [125]. Although various characteristics of primates have been analyzed in wildlife and lab experiments [113], many more aspects of their social life remain unknown [120]. The lifespan and reproduction cycle of primates can last up to 50 years. Most of the monitoring studies on primates have been conducted by either long hours of video recording or direct observation, and it remains a challenge for scientists to find out how they interact and how their social affiliations might change in the long run in their natural habitats.

Making use of the most recent advances in wireless networks can overcome many of the drawbacks and challenges in primate monitoring methods. Primates and their environment can be equipped with wireless sensor and actor nodes for continuous data collection. These nodes form a wireless sensor and actor network (WSAN) [1], in which sensor nodes sense events and a limited number of more powerful actors collect information from the sensor nodes. The actor nodes process this information and react accordingly. Recent improvements in size, weight, energy and sensing capabilities of sensor nodes as well as the self-organizing aspects of the participating nodes make WSANs suitable for wildlife monitoring [104, 106].

The evaluation of protocols using monitoring methods require efficient modeling of both the monitoring system and also the animal group under consideration. Therefore, realistic modeling of data collecting nodes and the behavior of primate network is critical. The initial formation of the animal network and the movement of the animal group must be modeled according to real-life observations and must reflect the social structure of the group. The realistic and long term movement data is missing for most of the primates [112, 113]. Consequently, it is crucial to use a suitable mobility model derived from the expected and observed mobility patterns.

The contributions of animal monitoring system are threefold. The first contribution is the monitoring system composed of wireless sensor and actor nodes, which are modeled in the OPNET simulation tool with a modular design. Then we approach the problem of the absence of realistic data to model the movement patterns from a social network perspective. The preferential attachment concept is used to introduce two network formation and mobility models for primate groups. There is no limitation in the node degree of existing preferential attachment models, which violates the known attachment limitations for different members in a group. Thus, the second contribution is the cut-off preferential attachment model based on spatial relationship among the nodes. This model is integrated with the Lévy walk mobility model [166] to define the foraging of primate groups. The third contribution is the role determination algorithm, which uses the collection of the spatial-temporal relationships to automatically and locally decide on the role of each animal in the society. The social network characteristics of the primate groups created by the model are verified by comparisons with the analyses conducted on real-life primate networks.

Natural extensions of our primate monitoring system would be the application of the system to human social networks. Therefore in this chapter we also consider the problem of generation of human social networks by using the interaction data such as physical proximity, text messages, phone calls and video chats. We start our approach to this problem by evaluating the interactions. The interactions are assigned weights according to their types. The total value of the interactions between two users is evaluated by using the number of communications and the weights of the interaction types. Since the interactions are quantified by our approach, these values are used to rate and rank the friends of users and to find the friendship levels in the social network of each user.

6.1 Animal Monitoring

6.1.1 System Model

The proposed animal monitoring system aims to collect data from a primate society by building a WSAN among nodes on apes and stationary nodes in the environment. For this purpose, sensor nodes are attached to apes in the network and a selected member of each group is equipped with an actor node. On particular locations in the habitat, additional actors are installed at accessible points to collect data from the actor nodes when they are in transmission range. These actors form the backbone of the network and work as the gateways of the mobile network.

Fig. 6.1 shows a possible implementation of the system for a primate society. The illustrated system captures and records movement and social interactions among primates continuously as the individuals move in their natural social life.

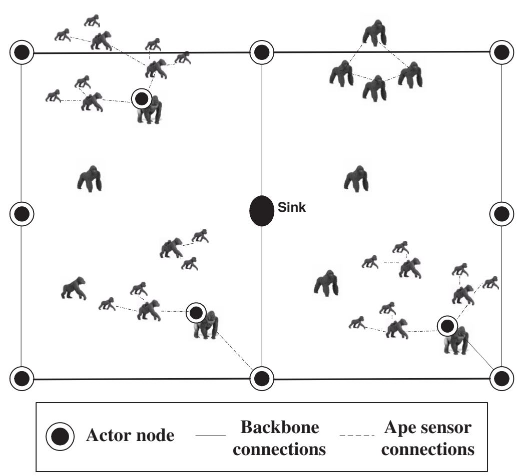


Figure 6.1: Application scenario of primates.

In our system model, each element has a transmission range r with a total transmission area of $\pi \cdot r^2$. The sensor nodes and actors are assumed to have maximum transmission ranges r_s and r_a, respectively, with circular transmission areas, where $r_s < r_a$ due to better computation and communication capabilities of the actors. For communication between two nodes, a bidirectional connection must be established. Hence, a device s_1 must be in the transmission range of s_2, fulfilling $d(s_1, s_2) \leq r_s$.

Communication links may fail or disappear from the network caused by constraints of the environment such as obstacles or by social changes in the group such as a conflict between two individuals. Additionally, nodes are mobile with variable speed and directions. Thus, the neighborhood of a node changes over time. For each sensor node s in sensor set S, we assume a neighboring list $Neigh(s)$, the set of nodes that are directly connected to s, such that $\forall u \in Neigh(s), d(s, u) \leq r_s$. $Neigh(s)$ is built when the network is deployed and it is updated with a certain frequency f if it is not triggered by an event.

All communication in our model is locality preserving. Hence, every node is restricted to communicate with its current 1-hop neighbors (a sensor node or an actor). When communicating with sensor nodes and creating the affiliation areas, actor nodes use the same transmission range r_s as the sensor nodes. Each actor has a weight value k, which shows the maximum hop count in the network. In other words, k is the maximum hop distance from an actor to a leaf node and it can be arranged according to the requirements specific to the observed animal group. Each actor encodes the transmitted packets with its ID and weight value k.

Since the collected data is transmitted in one direction and only local information is used, the construction of our system model does not require any multi-hop control communication. Stationary actor nodes use their full transmission range when the network is initialized to create the backbone among actors and the sink. In our particular application scenario, actor nodes are positioned such that each actor node has at least one actor or sink in its transmission range. Otherwise, the sink would be required to receive the collected data through the sensor nodes, which would potentially create severe packet loss and delay conditions in the network. Actor nodes use their full transmission range when they have data to exchange, consolidate and transmit to the sink. Although actor nodes typically have more resources than sensor nodes, resource constraints still apply to both sensors and actor nodes. Thus, alternating the transmission range of actors extends the lifetime of the network. There are various actor positioning strategies in literature, which can be integrated with our approach to improve data collection, area coverage and energy savings.

6.1.2 Network formation and mobility

In this section, we introduce our network formation and mobility models for the primate groups. The behaviors of primate societies and their social networks show great variety. Primates have complex social lives with families, affections, and politics of their own. Denham [167] presents a topology of primate societies according to social behaviors of different populations. According to this topology, a model is created for relations among environmental factors and primate social organization to define different types of primate societies.

The environmental inputs used in Denham's model include multiple factors such as the space and resource allocation, social motivations and mating strategies. After analyzing the primates according to these factors, Denham specified three important parameters namely, food predictability, food density and anti-predator strategy. Categorization of primates according to these parameters results in eight possible groups.

Only five of the groups defined in Denham's model are observed in nature and we concentrate on the group with the highest number of species including baboons, macaques, langurs, howlers, gorillas and chimpanzees. This group is defined to be living in a low food predictability, high food density environment and having an active anti-predator strategy. The animals in this group stay close to each other while foraging and they are structured either as one-male-several-female or multimale-multifemale groups [167]. The information and assumptions used in our approach follow the general guidelines about this group. Accordingly, the members of the group have different roles depending on the gender, age, strength and affinity. These roles are listed as follows:

- *Alpha male*: Alpha male leads the group in daily travels and has exclusive breeding rights to the females. Generally there is one leader alpha male in each troop.

- *Adult female*: Adult females usually compete to stay close to the alpha male. Generally there are three or four adult females in each group.

- *Juvenile male*: Juvenile males tend to stay not very close to alpha males since the alpha male can see them as threats to its authority. Maturing males usually leave their family groups to establish either their own band or to join a bachelor group.

- *Juvenile female*: A juvenile female stays closer to the troop compared to juvenile males and may change family groups a number of times.

- *Newborn and infant*: A newborn forms a very close relationship to its mother, rarely straying more than a few steps from her side for three to four years.

The roles and characteristics of the primate group are critical inputs for the introduced network formation and mobility models. Different primate groups would exhibit different social structures. The presented models provide a base model that can be adapted accordingly.

6.1.2.1 Network formation

The initial distribution of nodes in the environment is important when modeling the structure of a society. We introduce two approaches for initial network formation. While both approaches utilize the social structure information of the primate group under consideration, the first approach is based on the preferential attachment method and the second approach uses the center of mass concept.

The result of the network formation according to preferential attachment approach is a *"scale-free"* network [168]. In scale-free networks, the distribution of the number of node connections follows a power law distribution [119].

The network formation is initialized by positioning two sensor nodes in the area such that they are in transmission range of each other. Let G_n be the resulting graph of the network when the n^{th} node (v_n) is added to the existing nodes in the network. When obtaining G_n from G_{n-1} according to preferential attachment, the probability of adding a link from v_n to v_i, $P(i)$, is proportional to the degree (d_i) of v_i.

If the network is deployed solely according to preferential attachment, the resulting network would be unrealistic in terms of the connections among the nodes. For instance, in a network of n animals the preferential attachment method may result with a couple of nodes having more than $\frac{n}{4}$ connections. This resulting structure may match only a few animal societies, which do not have a complex social structure.

The network formation model of our protocol extends the preferential attachment to be effectively used for animal societies. The social structure of most animals is clustered such that there are subgroups in the entire animal group. For example the alpha male in an ape society is generally accompanied by multiple females, which are surrounded by their offspring almost all the time. A new parameter, called *"maximum degree"* (d_{max}), is introduced to include these properties of the animal social structures.

The network is deployed according to the preferential attachment until one of the nodes has the *"maximum degree"*. When a node has a degree of d_{max}, it becomes ineligible as a new deployed node for others to get connected. Therefore, the $P(i)$ value of a node depends not only on its degree but also on the d_{max} defined for the animal group. After the node reaches d_{max}, the node's $P(i)$ is reduced to the value P_c, based on the characteristics of the animal society. Hence the probability of adding a link from v_n to v_i is defined as follows:

$$P(i) = \begin{cases} \frac{d_n}{\sum_{i=1}^{N} d_i} & \text{if } d_i < d_{max} \\ \\ P_c & \text{if } d_i \geq d_{max} \end{cases} \tag{6.1}$$

where d_i is the degree of the node i. The decision process on the deployment of a new node joining the network is given in Algorithm 5.

In our application scenario, we consider multi-male multi-female structure with one group leader. Each ape in the network can be in the group of only one alpha male and there cannot be a link between two alpha males. Hence the links, which are against these rules, are removed as the networks are formed. The preferential attachment based network formation method is extendable by adding more species-specific features. For instance, as an ape group moves in its environment, the leader of the group avoids close encounters with other groups. Therefore the links between groups with separate alpha males are removed after the nodes are deployed and the roles are assigned.

Algorithm 5 Deployment of a new node

```
S = ∑_1^N d_i
r_s = Random number between 0 and S
d_a = Degree of node a
N_l = Number of leaders
N_max = Maximum number of leaders
for Each node a in the area do
    if (d_a > r_s) & (d_a = d_max − 1) then
        Connect the new node
        if N_l < N_max then
            N_l = N_l + 1
        end if
    else if d_a > r_s then
        Connect the new node
    end if
    r_s = r_s − d_a
end for
if The node not deployed then
    a is a solitary node
end if
```

Fig. 6.2 shows an example sequence for network formation. In this scenario, d_{max} and P_c are taken as five and zero respectively. Hence the nodes having a degree of d_{max} during deployment are no longer candidates for new nodes to get connected. In Fig. 6.2(e), the roles assigned to the nodes are shown. There are two alpha males in the society and it can be seen in the previous frame that they are connected by a link (thicker line). This link is removed after the roles are assigned according to the species specific rules under consideration, observing that there cannot be a link between two alpha males.

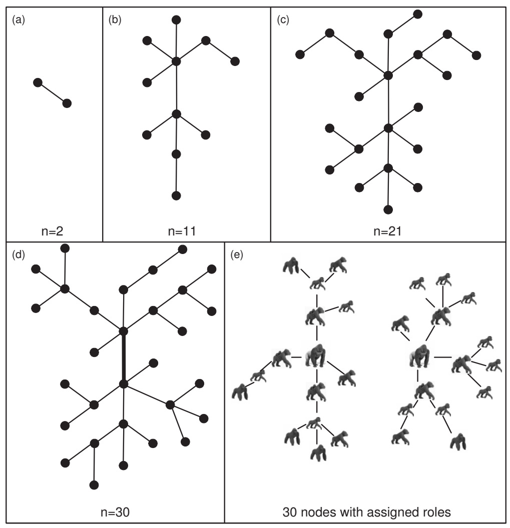


Figure 6.2: Deployment of nodes by preferential attachment based method.

In center of mass based approach, the nodes are distributed in the area according to a predefined structure. This structure depends on the type of the species under observation and the distribution of roles in this species' social network.

The center of mass concept is used in accordance with the hierarchy in the animal society. The animal society is divided into subgroups such that each subgroup's center of mass is their leader from the higher level group. For instance, the breeding females form a group and their center of mass is taken as the alpha male. Similarly, a mother ape is chosen as the center of mass for its offspring. This method can also be applied to other animal groups with hierarchical social structures. According to the center of mass approach, the coordinates of the nodes around the leader must satisfy the following equations:

$$x_s = \sum_{i=1}^{N} \frac{x_i}{N} \qquad\qquad y_s = \sum_{i=1}^{N} \frac{y_i}{N} \tag{6.2}$$

where (x_s, y_s) is the position of the leader and N is the number of nodes in that subgroup. This method is extendable for different scenarios. For instance if a mother has four infants, the possible positions of these nodes can be limited depending on their ages so that two of them will be very close to the mother whereas the others keep a larger distance from her.

6.1.2.2 Mobility model

Lévy walk is observed as the mobility model in most of the animal foraging patterns, such as jackals [169] or spider monkeys [170] and it is recognized as an optimal way to find randomly dispersed objects [171]. It is a random walk with step-lengths distributed according to a heavy-tailed probability distribution. Lévy walks are Markov processes and after a large number of steps, the distance from the origin of the random walk tends to reach stable distribution. The Lévy distribution is the Fourier transformation of the moving distance of a single random walk and Rhee et al. [172] gives its PDF as follows:

$$f_{z,\alpha}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-izt} \phi(t) dt \tag{6.3}$$

where $\phi(t) = e^{-|Ct|^{\alpha}}$ and C is a constant.

The distribution can be approximated by a power law of the form $y = x^{-\alpha}$ where $0 < \alpha < 2$. Each step in Lévy walk can be expressed by a tuple $L = (1, \theta, \Delta t_f, \Delta t_p)$. Δt_f indicates the walking duration and it is chosen for each walk from a probability distribution $P(l)$. Δt_p specifies pause time at the end of a walk and θ is the random direction taken by a node. A Lévy walk contains many short walks and a small number of long walks. The resulting pattern depends heavily on the value of α used in the system. As α becomes greater, the number of short walks increases.

One of the most common behaviors observed in diverse species is that they live in groups and follow the leaders of their groups. Scientists have various explanations such as the increased safety or breeding opportunities in a group for these behaviors. In nature, the alpha male makes the decisions for the selection of paths that the group follows. The alpha male role in a group is assigned to a node in the network formation phase of our scenario. Similarly, the mobility model of the group depends on the movements of the alpha male in both of the mobility models introduced. The alpha male moves according to Lévy walk and the movements of other members of the group are directed mainly by the alpha male's path. This path is used as the main input when positioning the nodes at each time instant. Two additional methods supplementary to Lévy walk are used to determine the mobility of each node.

Preferential attachment based method complements the preferential attachment based formation. After the nodes are deployed, based on the moves of the nodes leading their corresponding groups, the moving directions of the other nodes are probabilistically decided. The probability is defined according to the nodes' roles and levels of proximity to their group leaders. Since the deployment attributes of nodes such as their assigned degrees and roles determine their initial positions, these attributes and therefore the preferential attachment method also affect the decisions on the movements of the nodes.

Each animal moves based on the mobility of its neighbor with highest degree, its distance to this neighbor and this neighbor's moving direction. This is a characteristic of the mobility model, which matches with the hierarchical structure of animal swarms. Hence the mobility of the offspring is based on the mobility of its mother, whose movement in turn depends on the mobility of the alpha male. This structure also provides consistency with the approach of utilizing Lévy walk for the alpha male in order to obtain a Lévy walk pattern for the whole group.

When the group moves the animals close to their leaders tend to stay close to the same position relative to such leaders. For instance, the newborns or infants are generally at most only a few steps away from their mothers. However, juvenile animals forage in the environment and may walk in other directions. As they become adults, they may leave the group. In order to include these characteristics, the nodes in close proximity of their leaders follow the leaders with a high probability, which decreases slowly as the distance of the node to the leader increases. Consequently, the model provides a Lévy walk pattern to the group while providing possibilities for rare behaviors such as a bachelor male group formation. The probability $P_m(i)$ of a node to move in the same direction with its highest degree neighbor with transmission range r_a is as follows:

$$P_m(i) = \sqrt{\frac{r_a - d_i}{r_a} + c_1} + c_2 \tag{6.4}$$

where d_i is the distance between the nodes. The constants c_1 and c_2 are included to provide the functionality of adjusting the probability calculation for different species types or network requirements. The Fig. 6.3 demonstrates an example of two females 1 and 2 with their corresponding probability values $P_m(1)$ and $P_m(2)$ to move in the same direction with the alpha male, depending on their distances to the alpha male. They move in any other random direction with probability $1 - P_m(i)$.

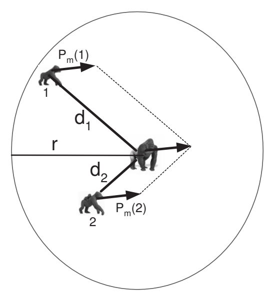


Figure 6.3: An example showing moving probabilities for females.

The ape scenario includes an additional feature such that the nodes from different groups repel each other such that the groups are physically separated in the environment.

Center of mass based method is used to determine the positions of all animals at each time instant after the network formation according to center of mass approach and the determination of the main path of the group. A node's neighbors at hierarchically one lower level move in a coordinated fashion to have that node's position as their center of mass at all times.

This method allows a more controlled mobile network compared to the preferential attachment based method since the hierarchical structure of the system remains in its initial format throughout the network lifetime.

6.1.3 Data Collection and Interpretation

6.1.3.1 Network structure

The data collection network considered in our approach is composed of actor and sensor nodes. The actors distributed in the environment are 'stationary' whereas the actors attached to the primates are considered as 'mobile' actors since the primates move in the environment. The primates equipped with the actors are selected according to their roles in the social structure of the group. The alpha males lead the primate group and they are accompanied by adult females, which have continuous connections to the young members of the group. Therefore we consider the alpha males as the cluster heads and equip them with actors. All the other members of the group are equipped with sensor nodes.

The network structure is formed and maintained by the state-transition rules defined only by local information. The nodes rapidly update their attributes as the network structure changes. A sensor node keeps a maximum weight value for each actor it receives packets from and it does not build a state or history of the whole network. The sensor nodes are affiliated with both stationary and mobile actors in the network.

When a sensor node receives a packet from an actor, the packet is retransmitted if its weight is less than the sensor node's weight for that actor. Otherwise, the sensor node drops the packet to avoid unnecessary traffic and energy consumption in the network. The weight $W_A(s)$ of each affiliation corresponds to k-hop distance of a node to an actor. The condition, in which a sensor node does not receive any weight updates, is defined as the loss of connection for the sensor node, which may refer to a solitary animal. In that case, the sensor node sets its hop value to the minimum value defined for the network. Then it operates only in 'listening' mode and does not transmit any packets.

Algorithm 6 utilizes the spatial proximity between two nodes to decide on the weights of the sensor nodes. The weight information for the nodes in a group is collected at the corresponding mobile actor and transferred to the backbone whenever it is possible. The periods of time for specific weight values are also calculated and updated in Algorithm 6 to be employed by actors as follows:

$$p_{w_a(s)_{last}} = p_{w_a(s)_{last}} + t_{current} - t_{w_a(s)_{last}changed} \tag{6.5}$$

where p_v is the time period for a node's weight value v and $w_a(s)$ is the weight of node s for actor a.

Algorithm 6 The state transitions of a node s

$w_a(s)$: The weight of node s for actor a

p_v: The time period for a node's weight value v

$max(w(Neigh(v_s)))$: M

if Received a local update **then**

  if s is not affiliated with an actor node **then**

    $w_a(s) = 0$

  else if $M = k$ **then**

    $w_a(s) = k - 1$

  else if $(M! = k)\&(M > w_a(s))$ **then**

    $w_a(s) = M - 1$

  else if $M < w_a(s)$ **then**

    $w_a(s) = w_a(s) - 1$

  else if $Neigh(v_a) = O$ **then**

    $w_a(s) = 0$

  end if

  if $w_a(s)_{last} \neq w_a(s)$ **then**

    Update $p_{w_a(s)_{last}}$

    $t_{w_a(s)changed} = t_{current}$

  end if

end if

Information about a group can be collected by more than one actor. All the collected data are integrated at the sink to decide on the social structure of the ape population.

6.1.3.2 Role determination

Animals in a primate group have different roles depending on the gender, age, strength and affinity. Algorithm 7 gives the basic role determination process and the utilized rules. The transmission range, time and cardinality of the actor affiliation group are the parameters defining the rules for role assignment of the animals. The thresholds of these values can be generalized according to the animal group under observation while the rules are very specific for the social network of primates.

Algorithm 7 Determination of roles

$w_m(s)$ =Weight of node s for actor m

k =Actor weight

t_{hp} = Time for high proximity

t_{lp} = Time for low proximity

if $w_m(s) \neq 0$ **then**

    if $w_m(s) = k - 1 > t_{hp} \& w_m(Neigh) = w_m(s) - 1$ **then**

        s is on a Female

    else if $t(w_m(s) = k - 1) > t_{lp}$ **then**

        if $t(w_m(s) = k - 2) > t_{hp}$ **then**

            s is on an Infant

        else

            s is on a Juvenile female

        end if

    else

        s is on a Juvenile male

    end if

else if $Neigh(s) \neq 0$ **then**

    s is on a bachelor group male

else

    s is on a solitary animal

end if

Algorithm 7 uses the basic roles defined for all primate communities we focus on. The individuals closest to the alpha male are the adult females. If a primate is determined as an adult female with offspring, the distance analysis is executed with varying transmission range in order to check the distances of the offspring. This is used as a method to decide on the ages of the primates since infants stay close to the mother ape for most of the day as well as when they are sleeping. Additionally, this analysis helps to decide on the data aggregation points in the network since mothers form perfect data aggregation points for the data gathered from the offspring. Same type of a distance analysis can also be used for juvenile male primates since generally as far a juvenile primate keeps himself from a group, as it is likely for him to leave the group.

The feedback from the system is used to make new decisions. As the mobile actors collect information from the network and bring it to the backbone, the sink will update the roles of the apes depending on the feedback from the network. If the feedback shows that a group of juvenile males form a bachelor group, one of these animals is chosen as the data collection gateway for the group. This is efficient in terms of data collection and energy efficiency.

The determined roles of the nodes in the network must be combined with the social network analysis methods to understand the characteristics of primate networks. Studies on primate societies show the importance of social network analyses and different aspects of network statistics [125] on understanding and identifying social structures of primates. One of the most commonly used metrics in these studies is the clustering coefficient [124], used to measure the extent to which vertices adjacent to any vertex v are also adjacent to each other [173]. Another commonly used metric is the eigenvector centrality, which measures the influence of nodes in the network by assigning relative scores to all nodes based on the the number of edges such that a connection to a high-scoring node contributes more to the score of a node than a connection to a low-scoring node.

6.2 Ego network generation based on mobile phone data

In this section, inspired by our approach in primate monitoring system, we deal with the problem of automatically generating the social network of a person by using different sources of available interaction data such as physical proximity, text messages, phone calls and video chats. The rating and friendship level of a user's friend is determined by our approach as shown in Fig. 6.4.

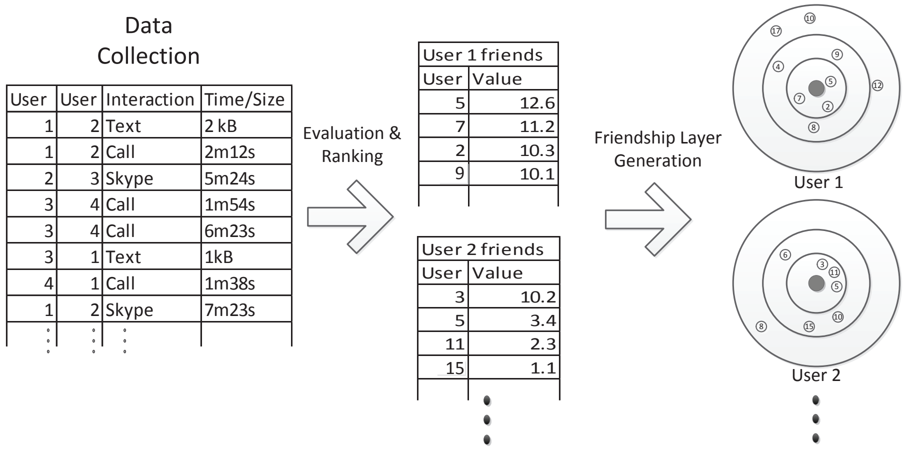


Figure 6.4: Overview of social network generation for users.

There are various applications on smart phones, which allow us to trace multiple types of interactions between a user and the members of the user's social network. For instance, text messages, call logs and e-mail conversations are stored as history on the device or in the cloud. Moreover, smartphones are equipped with many sensors, which sense, evaluate and record even more information about the user, the environment and contacts. For instance, the proximity of two users is detected by acoustic sensors or by applications using Bluetooth.

Most of the mobile phones are equipped with GPS receivers where the location information can also be collected through them.

Usage of all available data on a mobile phone enables the interpretation of the social network. However, one of the critical problems is that the generation of the network cannot be measured in terms of accuracy or other measures that would allow us to state that the collected data sufficiently describes the corresponding social network. The answers of the questions related to social network dynamics are visible influences on models such as ours.

Our ego network generation approach is applied to a reallife dataset, which is collected during the Nodobo project [1]. Nodobo is an experiment to gather communications metadata from a group of high school students. For this purpose, each student is provided with an Android cell phone. The collected interaction data of Nodobo project is analyzed in terms of basic interaction information such as the identifications of the parties in the communication, the time of the communication and type of the communication method. Instead of plainly reading and transforming the interaction data, these structural artifacts are used as a structural framework. Additional to the information derived from the interaction data, this framework gives information on how the nodes are socially located in the network and how the links are created.

6.2.1 Interaction evaluation

The interaction data derived from mobile devices and sensors need to be evaluated in terms of their importance. The type of an interaction is an important factor to determine its importance. Intuitively, an e-mail sent to a person appears to be more distant than calling the same person. A phone call is shown to be less effective for personal relationships than meeting with the other party in person [174]. Considering these differences, we suggest assigning specific weights to different types of interactions. Weight assignment results in the ability to change the weight according to experience or context and to include additional interaction types as needed. The interaction values are defined in our approach as follows:

$$i_{A,B} = \alpha \cdot F(T) + \beta \cdot V(T) + \gamma \cdot nP(T) + \delta \cdot E(S) \tag{6.6}$$

where $P(T)$, $V(T)$ and $F(T)$ denote the number of times a phone call, a video conference and a face-to-face interaction occurred respectively for a particular amount of time, T. $E(S)$ denotes the number of e-mails or text messages with size S. The number and types of interactions can be increased or decreased based on the capabilities of the mobile phone or the application to collect the data.

Each interaction type has a different constant $(\alpha, \beta, \gamma, \delta)$, which reflects the variety in effects of different interaction types on personal relations. Formulation of this equation and finding the exact values of constants is one of the next steps of our work. Due to the nature of social sciences, these values may change depending on various factors such as the social group under investigation. Our approach provides the means to utilize results of works in social sciences such as the study of Okdie et al. [175].

6.2.2 Ranking Friends

The interactions between friends correlate in number with the strength of friendship according to Social Brain Hypothesis (SBH) [174]. Therefore, after determining the friends of a person and evaluating the interactions, our system also ranks the friends to find different levels of friendship in the social network of users.

6.2.2.1 Colley method based ranking

Similar to our approach in chapter 3, we define a sports competition style relationship among users. The friend with a larger interaction value in a defined period of time has a win against the friend with lower interaction value for the same time period. Therefore the sports ranking methods, in which teams win or lose against each other, can be utilized to rank friends.

The most common sports ranking method utilizes the winning percentage to rank the teams or individuals participating in the competition. The winning percentage is the ratio of the games a team won to the total number of games played by that team. Hence this method can be employed to find the rating of a friend i of a user u by using the following formula:

$$r_i = \frac{w_i}{(w_i + l_i)} \tag{6.7}$$

where w_i and l_i are the number of wins and losses of the friend i of user u.

When the Winning Percentage method is used to calculate the ratings of friends, the rating of a friend depends only on the number of times that friend has a better interaction value than the other friends and the number of comparisons made. Win percentage generally does not satisfy the particularities of sports organizations. Therefore, more complicated ranking methods such as Colley [65] and Massey [66] are used in sports.

According to Chartier et al. [69], the insensitivity is a desirable property for a ranking method in the social network analysis. For instance, when a person moves to another state or starts to work in a new company, the social network of the user may be extended by new friends. Then, the user may spend most of the time with one of these friends. If a sensitive ranking algorithm is used, the new friend immediately becomes a strong tie of that person, which rarely happens in real social networks.

Similar to our approach in chapter 3, we use the Colley method as the basis for ranking friends of a person. The Colley method of sports ranking can be defined by a linear system [156], $C\vec{r} = \vec{b}$, where $\vec{r}_{n\times1}$ is a column-vector of all the ratings, $\vec{r_i}$. The right-hand-side vector, $\vec{b}$, is defined with the components as follows:

$$b_i = 1 + (w_i - l_i)/2 \tag{6.8}$$

$C_{n \times n}$ is called the Colley coefficient matrix and defined as follows:

$$C_{ij} = \begin{cases} 2 + (w_i + l_i) & i = j \\ \\ -n_{ij} & i \neq j \end{cases} \tag{6.9}$$

The scalar n_{ij} is the number of times friends i and j are compared to each other. The Colley system $C\vec{r} = \vec{b}$ always has a unique solution since $C_{n \times n}$ is invertible. Then the rating of a friend of a user is defined as follows [69]:

$$r_i = \frac{1 + \frac{(w_i - l_i)}{2} + \sum\limits_{k \in F_u} r_k}{2 + (w_i + l_i)} \tag{6.10}$$

where F_u is the set of all contacts that user u communicated.

With this calculation method, the rating of a friend depends on the ratings of the other friends of user u, which has an important reflection in the social networks. Hence interacting with someone more than that person's best (top ranked) friend has a high impact on the friend rating. In contrast to traditional methods, the initial rating of any friend with no changes is equal to $\frac{1}{2}$, which is the median value between 0 and 1. Depending on the comparisons, a win increases and a loss reduces the value of r. This approach results in a system less sensitive to changes. Therefore the communication between a user and a friend needs to retain a high level for several ranking periods until it has a remarkable effect on the friend's rating. In other words, a friend of a user is not assigned with a high intimacy role by Colley method by just having a high interaction value for a short amount time.

After their ratings are calculated, the friends of a user can be sorted and assigned with ranks. Generation of a complete list of friends for a user and their rankings depend on the level of variety in the collected data, the weights assigned to different interaction types and the length of the interval for the data collection. The weights and the length of the interval must be chosen according to the sociological characteristics of the group under consideration. The ratings calculated by the ranking method used in our approach depend on the ratings of other users and the ratings are updated periodically. Therefore, the resulting friend network for a user will have a pattern of discrete groups of ratings according to Dunbar et al. [174]. Then, the levels of the intimacy or the circles of friends are decided according to the ratings.

6.2.2.2 Trust based ranking

There is a strong correlation between the level of trust and the intimacy in a friendship [176]. Sutcliffe et al. [177] offered a trust based simulation model for the development of social relationships, which reflects the social structure presented by SBH. This model focuses on ego networks rather than complete social networks and bases the formation and development of social relationships mainly on the interactions and a dynamically adjusted trust level between people. In particular, frequency and history of interactions, current level of trust and friendship maintaining strategy of people are defined as the factors determining the intimacy levels of friendships in an ego network.

According to Sutcliffe et al. [177], interactions may have positive or negative results depending on whether they are collaborative or defective. Besides being one of the major influences defining the intimacy level of friendships, trust also facilitates collaboration. Hence the trust accumulates for collaborative interactions and weakens by defective interactions. The change in trust is controlled by smoothing algorithms to implement distinct behaviors of high-trust and low-trust relationships. Therefore accumulation or weakening rate and amount of interactions are determined by the current level of trust.

To reflect influences on trust levels, Sutcliffe et al. [177] used the following model parameters: defect/cooperation rates for friends, waning rate of trust and linear/log functions for trust increase and decay. Initially trust level between the ego and a friend increases linearly according to the number of cooperative interactions. The rate of increase decreases progressively as the value of trust increases. Therefore, log functions are applied for trust levels instead of linear functions to reflect this behavior. Since the log function decreases the rate both for positive and negative interactions, defections affect high-trust relationships less compared to the low-trust relationships. For instance, a single non-collaborative interaction with a best friend does not substantially affect the trust for that person whereas it may seriously reduce trust in a new friendship. Additionally, there is a constant slow waning rate for trust in relationships, which is independent of any interactions.

The model by Sutcliffe et al. [177] is proposed to simulate ego networks. This model uses the interactions between two people to determine the trust level. Although an interaction is defined in the model to be either collaborative or defective with a certain probability, the model does not include a differentiation method for different types of interactions. We utilize its principles with integrations to our interaction evaluation method to determine ego network characteristics of users in a given dataset. According to the results of the model proposed by Sutcliffe et al. [177], social interaction strategies which favor interacting with existing strong ties results better than strategies favoring more relationships with weak ties. These strategies result in ego networks in compliance with SBH model.

Trust levels between individuals are built as they interact with each other. Fig. 6.5 summarizes the underlying process for calculating the trust level in a relationship. The total data collection time is divided into equal portions. At each of these portions of time, if there is an interaction between two users, the weight of interaction is applied according to its type. Then the interaction is probabilistically classified on whether it is collaborative or not. If the interaction is collaborative, the trust level between two individuals increase by the rate of choosen method. Otherwise it is decreased by the same amount. In case of no interactions, trust level decreases with a constant rate.

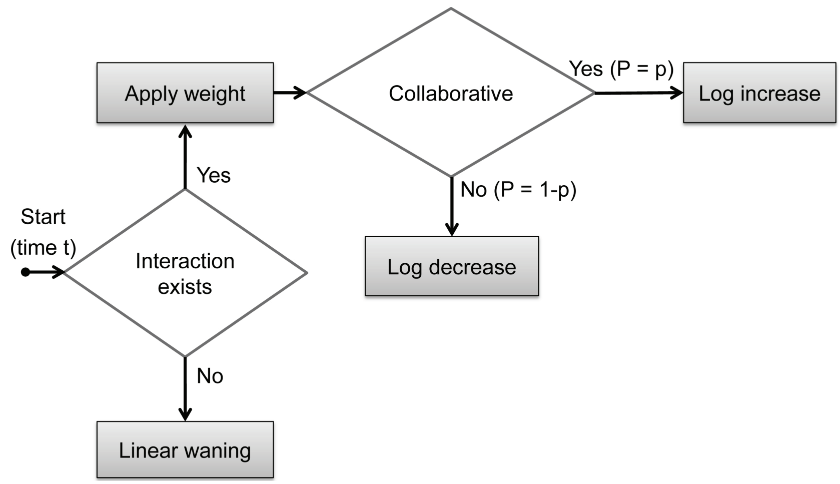


Figure 6.5: System to change trust level of a relationship in a time instant.

Besides interaction prioritization, following three parameters are important to generate trust levels for the process summarized in Fig. 6.5:

- Probability of an interaction to be collaborative (p).
- The rate of increase or decrease in trust with an interaction.
- Definition of strong, medium and weak ties.

Based on the model of Sutcliffe et al. [177] and SBH, logarithmic increase rate and linear decrease rate give the best matching result with SBH distribution of ties. The probability of collaborative interactions must be greater than 90% not to have an artificially high drop in the trust levels. Trust distribution for friends of a person is divided in three main levels. Strong ties are with the friends in the upper one third and weak ties are in the lower one third of this distribution. Medium ties fall in between strong and weak ties.

6.3 Simulation results

6.3.1 Animal monitoring

The performance analysis of our approach is carried out by extensive simulations in OPNET modeler [158]. The Lévy walk mobility model is used with $\alpha = 1.9$ and $\beta = 1.6$, which are the values for the foraging pattern of the considered gorilla troup based on the observed values in the nature [112, 170]. The segment-based trajectory modeling is used to create Lévy walk mobility. The trajectory consists of multiple points defined by Lévy walk with the given α and β values. Trajectories for Lévy walk mobility models are created and assigned to nodes as attribute values in OPNET. The random waypoint mobility model is also used in a group of simulations with speed uniformly distributed between three and eight km/h interval and pause time uniformly distributed between 50 and 150 seconds interval. The range of values for these metrics are determined by using the documented observations on gorillas [112].

The primate type gorilla is chosen for simulation studies. Gorillas live in socially organized groups, which are called as *"troops"*. The roles of the individual animals in the society build up a hierarchical structure, which is shown in Fig. 6.6 for our application scenario.

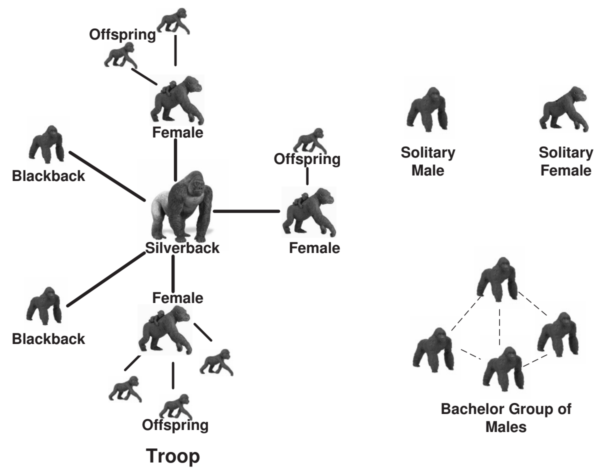


Figure 6.6: Gorillas live in socially organized groups.

6.3.1.1 Network formation

The proposed network formation model is tested against the original preferential attachment model in two sets of simulation studies. In the first set of simulations, the deployment of only a single troop is considered. Gorilla troop populations in nature usually range from 2 to 12 members and the average troop size is 9. The total of 50 simulation runs are executed and in each simulation, 11 nodes are deployed in the area. In Table 6.1, the average number of nodes for each number of connections are presented for the original preferential attachment and our preferential attachment based network formation and deployment model (PABD).

Table 6.1: Average number of nodes for each degree in 1^{st} scenario

Degree	0	1	2	3	4	5	6	7	8	9
Pref. Att.	0	7.03	1.83	0.93	0.46	0.30	0.21	0.18	0.13	0.09
PABD	0.23	6.30	2.27	0.73	0.53	0.83	0	0	0	0

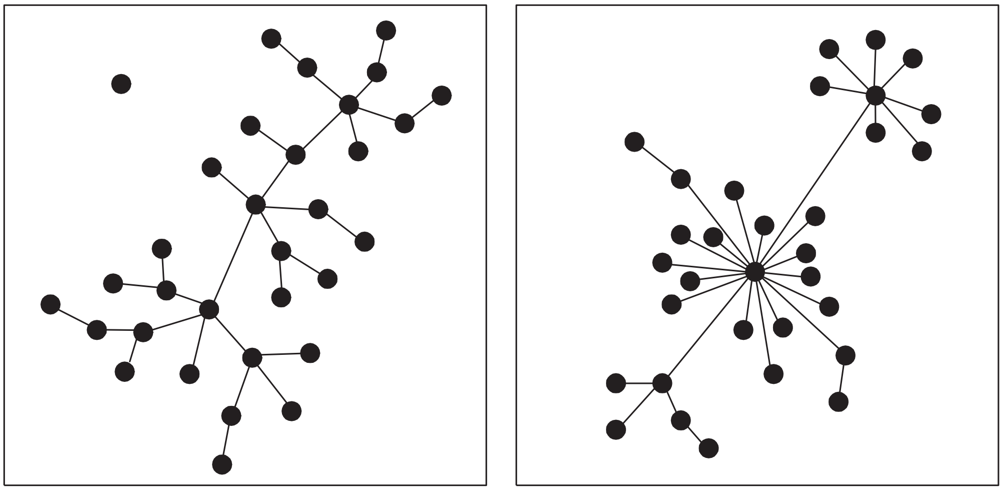
There are 32 nodes deployed in the area for each simulation of the second set of network formation experiments. Table 6.2 shows the average number of nodes for each number of connections for the original preferential attachment and PABD.

Table 6.2: Average number of animals in 2^{nd} deployment scenario

Degree	0	1	2	3	4	5	6	7	8	>8
Pref. Att.	0	20.33	5.97	2.11	1.20	0.60	0.47	0.33	0.20	0.79
PABD	0.66	17.45	5.17	2.93	2.03	3.03	0	0	0	0

The results given in Table 6.1 and Table 6.2 indicate that our network formation method produces gorilla troops with social relational properties similar to the troops in nature. The first result indicating this observation is that the degree distribution is more homogeneous compared to network formation by original preferential attachment while not any of the nodes exceeding a certain degree value. Another important result is that the solitary gorillas can be observed only in some scenarios of our network formation method.

One of the network formation cases by PABD, which has number of connections close to the average values, is presented in Fig. 6.7 to demonstrate the results visually. The figure shows that most of the nodes are directly connected to only a few nodes in the preferential attachment model, which is not a characteristic observed in a gorilla society [113].



(a) Deployment by PABD. (b) Deployment by Pref. Attachment.

Figure 6.7: Network formation by our protocol and by preferential attachment.

Fig. 6.8 and Fig. 6.9 show the distributions of node degrees for preferential attachment and our protocol in log scale. Even though the number of nodes is not very large, the power law linearization in a log scale is observed. This behavior is not observed in PABD as our network formation model extends the preferential attachment to be effectively used for animal societies. When a node has the maximum degree (d_{max}) defined for the group in consideration, it becomes ineligible for a new deployed node to get connected. Therefore, Fig. 6.8 and 6.9 show that the probability of adding a link to a node in PABD depends not only on its degree but also on the d_{max} defined for group in consideration.

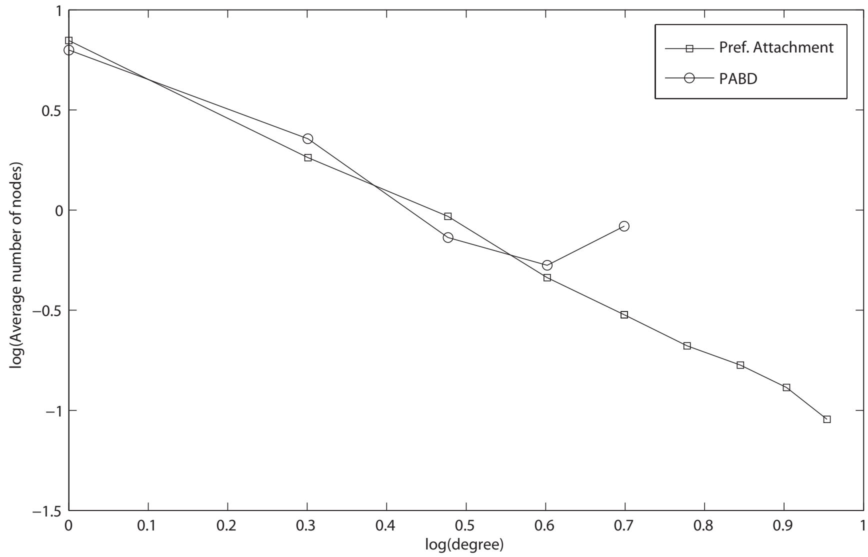


Figure 6.8: Degree distribution of 11 nodes for PABD and by Pref. attachment.

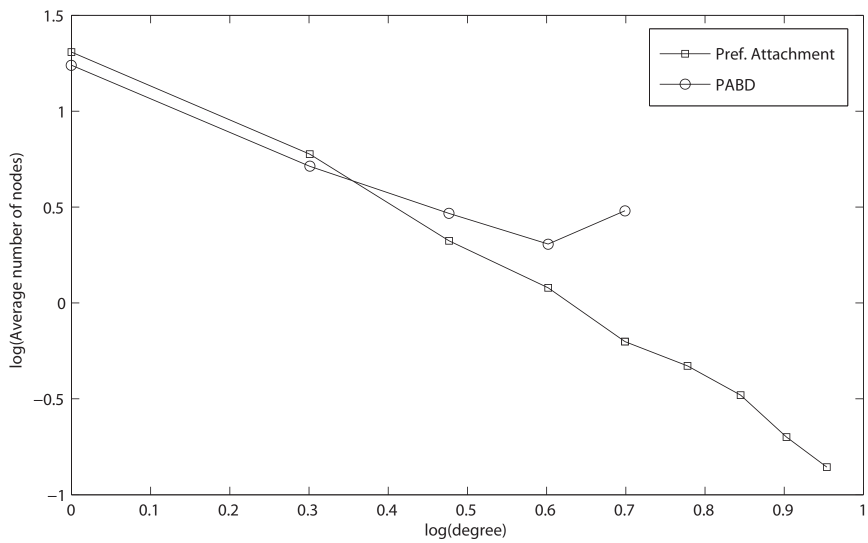


Figure 6.9: Degree distribution of 32 nodes for PABD and by Pref. attachment.

6.3.1.2 Roles

In the first set of experiments, nodes move according to the Random waypoint mobility pattern. Twenty simulation runs were executed with the same initial conditions and the average percentage of roles are determined by our role determination algorithm.

In Fig. 6.10, the percentages of the troop members over the simulation period are given. The percentages of the animals, which are not members of the troop, are given in Fig. 6.11.

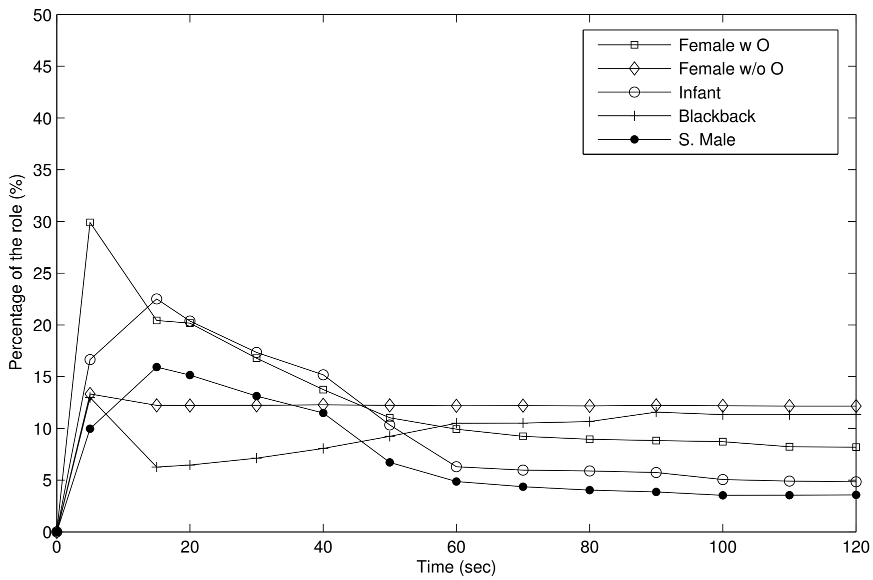


Figure 6.10: Members of the troop in random walk.

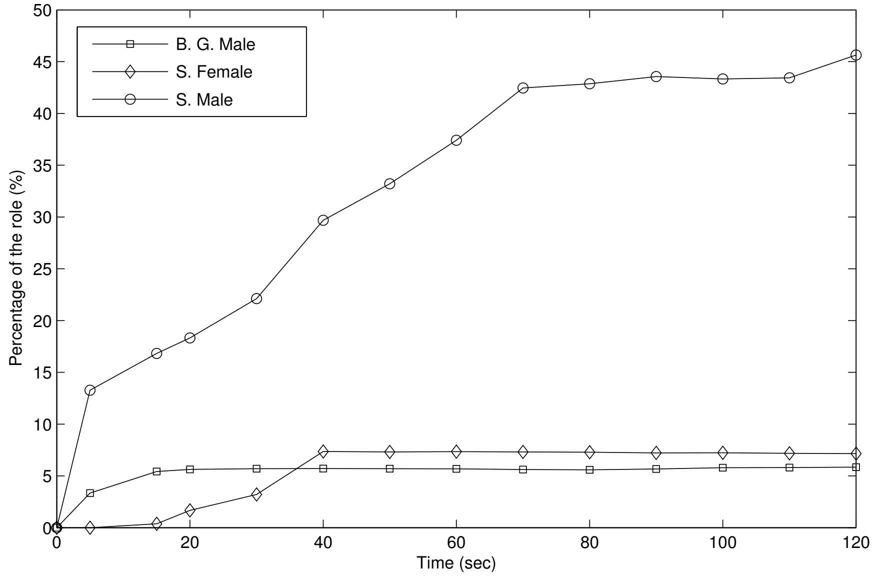


Figure 6.11: Non-members of the troop in random walk.

Fig. 6.10 and Fig. 6.11 show that the number of solitary males increase with the simulation time and this role clearly becomes dominant in the society. The other roles have similar shares, mostly depending on their initial conditions. The solitary males in the nature walk alone in the environment and they generally get affiliated with multiple troops over time. Hence this is an expected property for the society with random mobility with a starting condition in which most of the nodes are close to each other. Most of the nodes get departed from the troop as the time passes and the algorithms change their assigned roles to solitary males as they start to range alone in the area.

In the second set of experiments, the nodes move with Lévy walk with center of mass mobility pattern. The simulations are executed with the same conditions as in the initial set of experiments and the average percentage of roles are determined by our approach. The change in the percentage of the members and the non-members of the troop over simulation time are given in Fig. 6.12 and Fig. 6.13. Similar to the simulations with random walk, the percentage of each role fluctuates at the beginning of the simulation. However the fluctuation range is smaller and the fluctuation time is shorter compared to random walk. After the short fluctuation period, the roles in Levy walk become stable and the roles match the starting roles. In accordance with its design purpose, Lévy walk with center of mass mobility pattern provides an animal group with a stable role distribution.

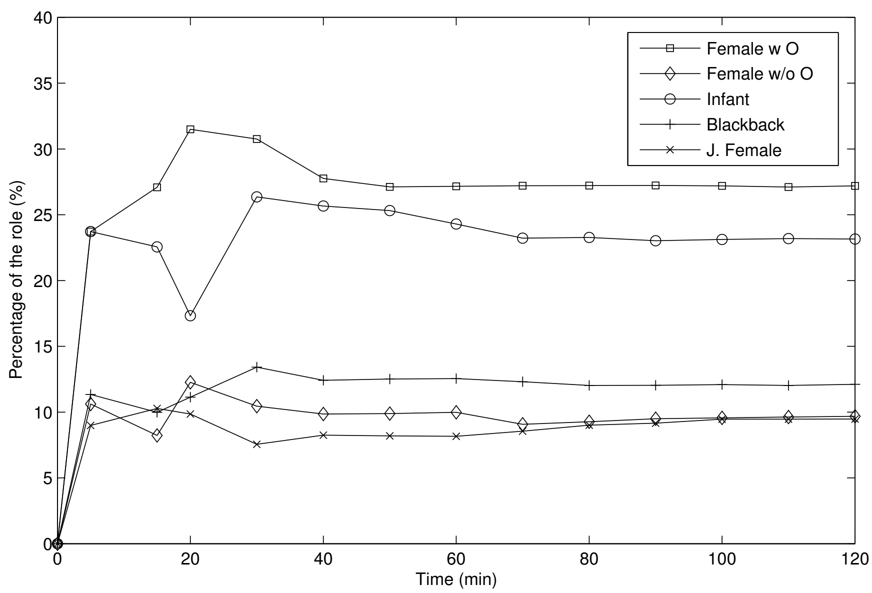


Figure 6.12: Members of the troop in Lévy walk.

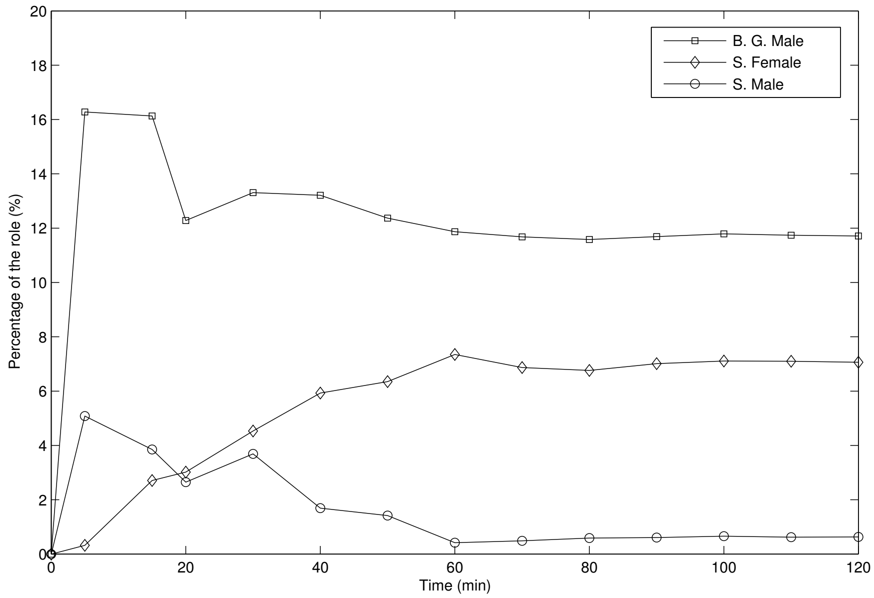


Figure 6.13: Non-members of the troop in Lévy walk.

In the third set of experiments on roles, the mobility of the nodes is defined by the Lévy walk with preferential attachment extension. The results of the simulation runs, which show the percentage of members and non-members of the troop over time are given in Fig. 6.12 and Fig. 6.13. The results show that the percentage of each role fluctuates at the beginning of the simulation similar to the initial experiment set with random mobility. This characteristic demonstrates that the role decision process requires a period of time to assign the correct roles to the animals. The experiment also shows that a Lévy walk with preferential attachment extension is an appropriate choice for mobility model, since the resulting average percentages of the roles match the observed structure of the gorilla societies [113].

6.3.1.3 Rules

The introduced protocol for social role determination can be applied to any different types of ape groups by modifying its rules or by creating new metrics. The graphs in Fig. 6.14 and Fig. 6.15 show results for two different metrics chosen for the simulation scenarios with our preferential attachment based mobility (PABM), center of mass based mobility (CMBM) and random walk (RW). For the first case, it is assumed that the role distribution of the mobile society must be same as the role distribution in the stationary case. The metric for Fig. 6.14 is the ratio of roles distributed in the mobile scenario to the roles distributed in the stationary case. For Fig. 6.15, the metric is the ratio of solitary animals to all animals in the society. These figures show that CMBM gives results according to its design purpose such that the cluster structure remains same as the initial conditions. The results also demonstrate the probabilistic nature of PABM since it differentiates from the initial deployment with a certain probability.

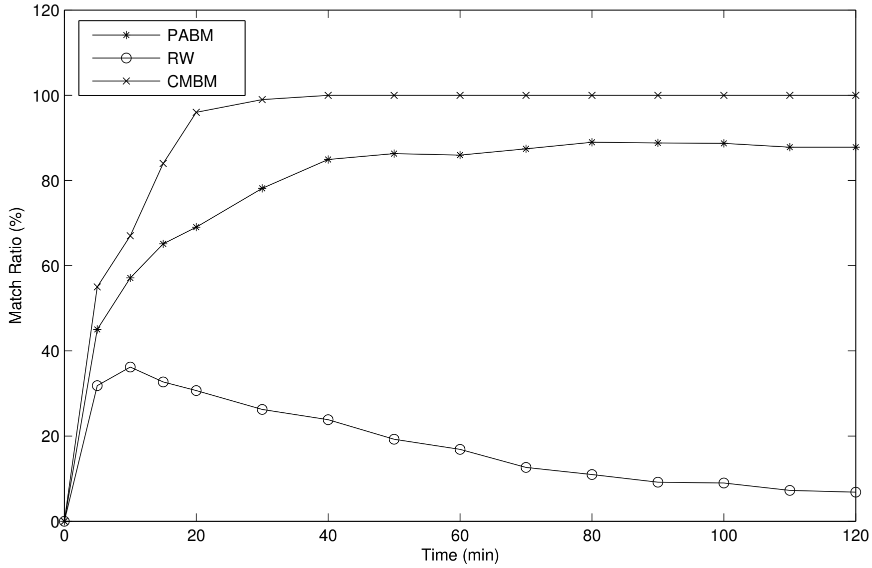


Figure 6.14: The ratio of roles in mobile scenario to stationary scenario.

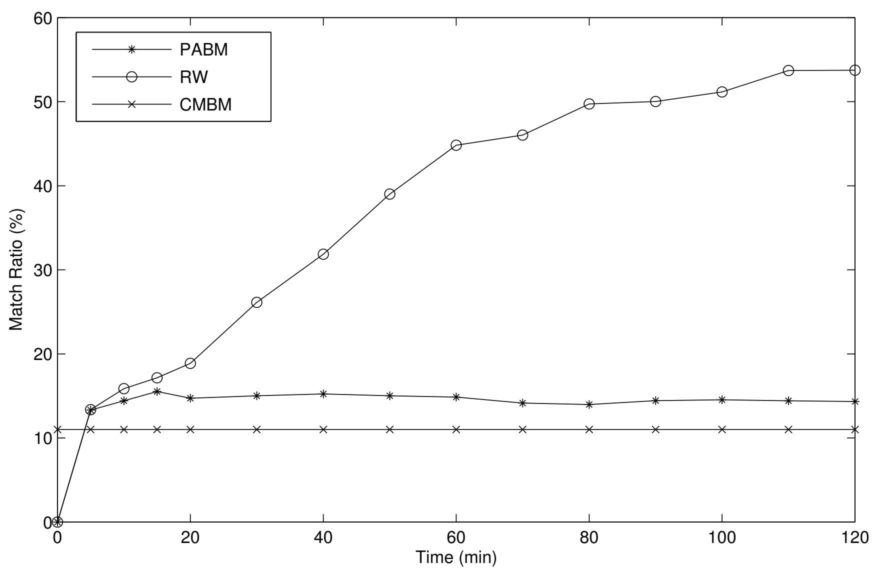


Figure 6.15: The ratio of solitary animals to all animals in the society.

6.3.1.4 Network analyses

In this set of simulations, we compared our approach to real-life primate networks using multiple social network metrics. We compare the characteristics of five real-life primate networks to the primate networks created by our approach. Two of these five groups are chimpanzees [125], two of them are macaques [123, 178] and one of them is a capuchin group [179].

Fig. 6.16 shows the clustering coefficient and eigenvector centrality values for real-life data and our simulations. The results show that our approach produces networks with similar characteristics to real-life primate social networks. The calculated clustering coefficient values are within the 10% variance of their mean and our simulation results are within the 5% variance of the mean. The values for eigenvector centrality are even closer to each other. The difference between the eigenvector centrality values of our approach and chimpanzee networks are less than 0.1%.

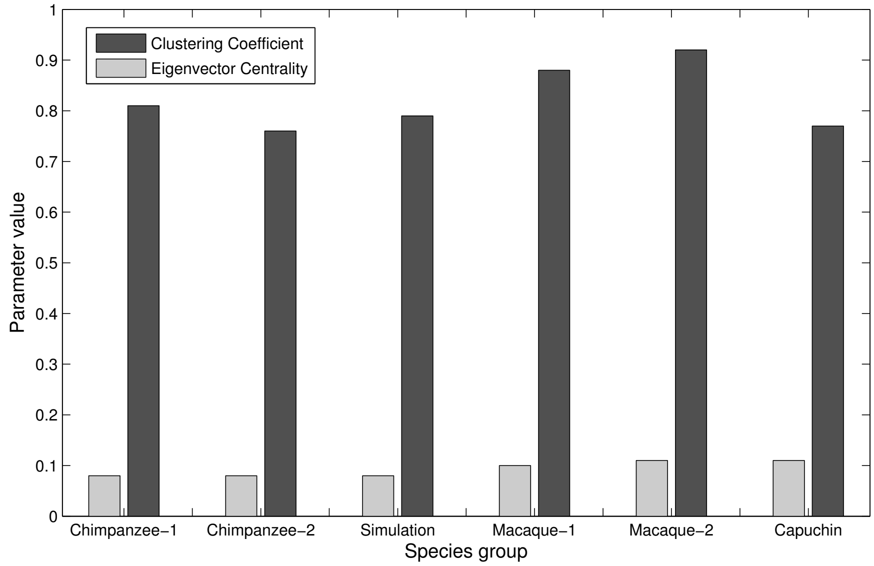


Figure 6.16: CC and eigenvector centrality values for real-life data and our simulations.

The proximity of the individuals throughout the time is used to create the network graphs of real-life primate data and our approach. When creating the graphs, the members of the group form the nodes of the network while the edges are drawn according to the time spent together. Edges are assigned with four different levels of weights to differentiate the strength of ties. Fig. 6.17 shows the graph of a chimpanzee social network [125] formed by real-life data and Fig. 6.18 shows the graph of a gorilla social network formed by our approach.

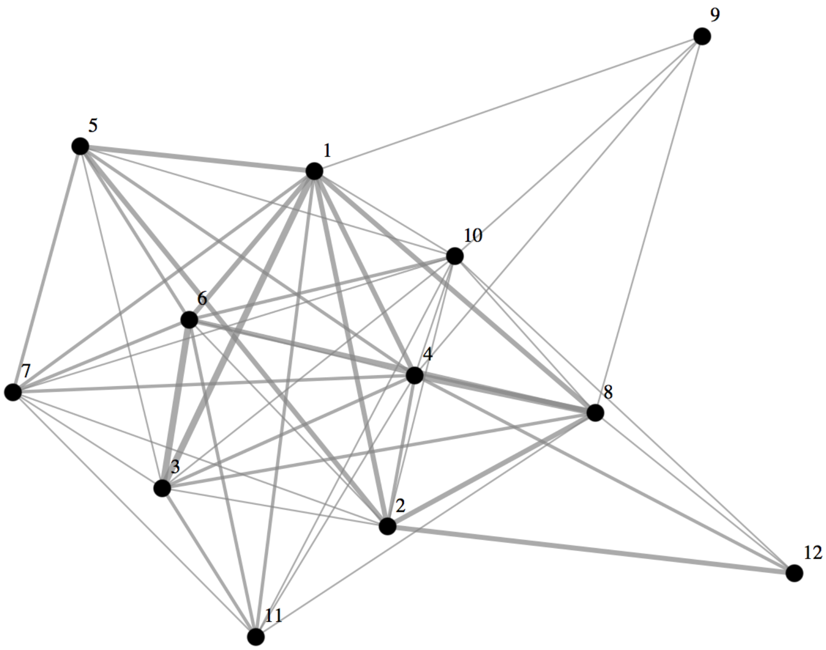


Figure 6.17: Network graph formed by using data of a real-life chimpanzee group (1).

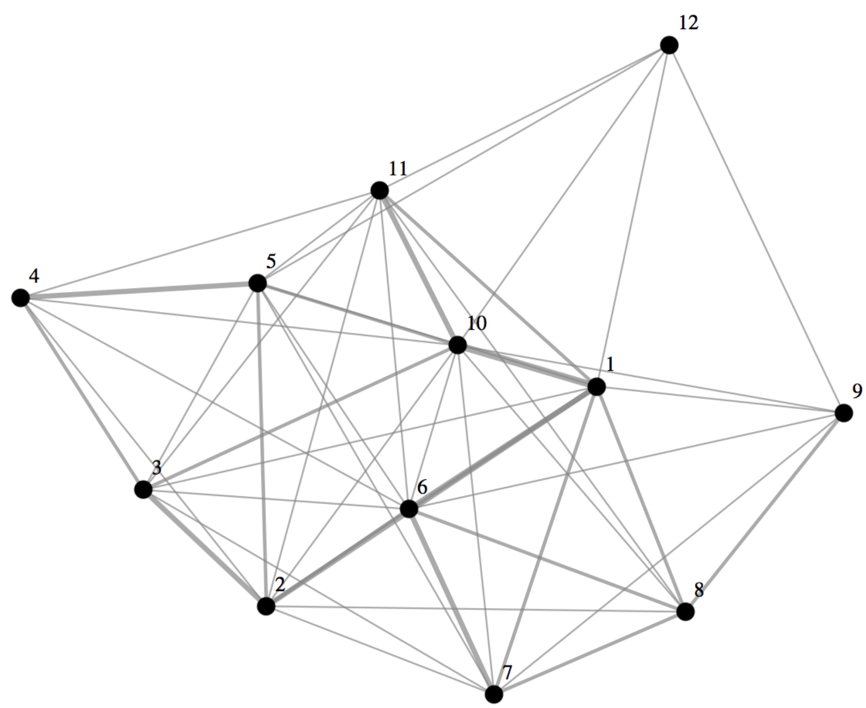


Figure 6.18: Network graph formed by using our simulations of a gorilla group.

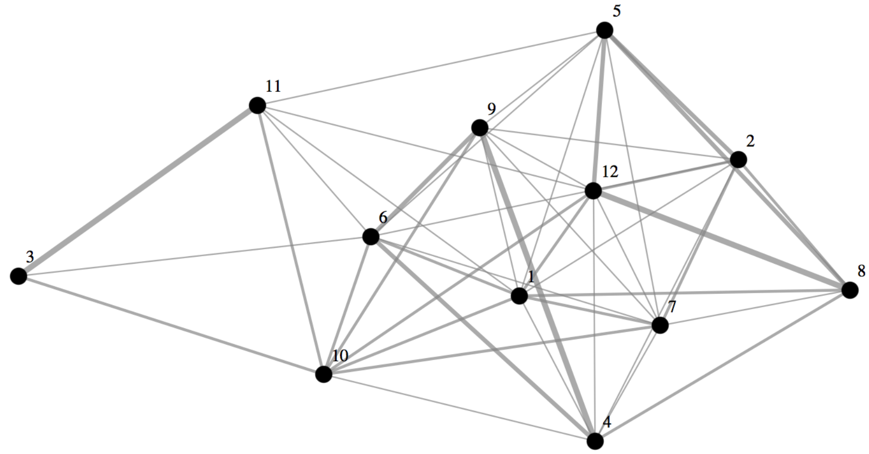


Figure 6.19: Network graph formed by using data of a real-life chimpanzee group (2).

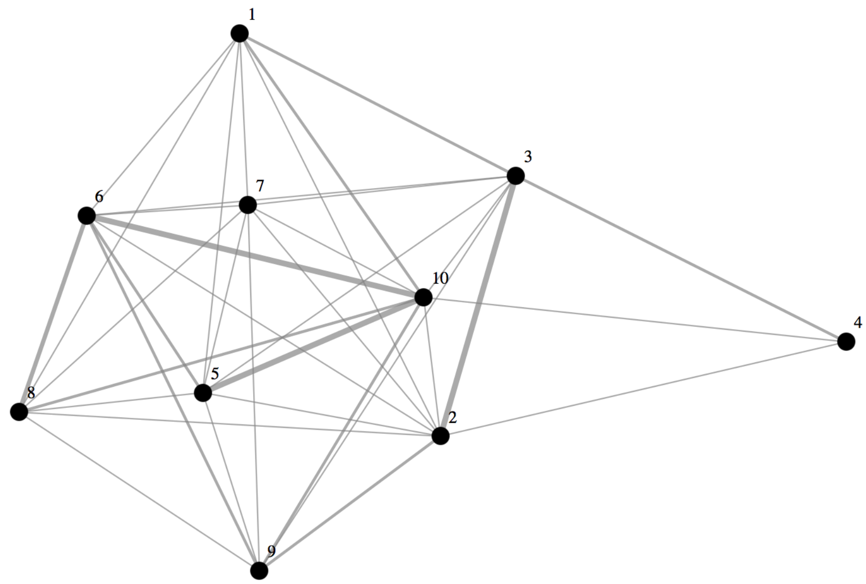


Figure 6.20: Network graph formed by using data of a real-life macaque group.

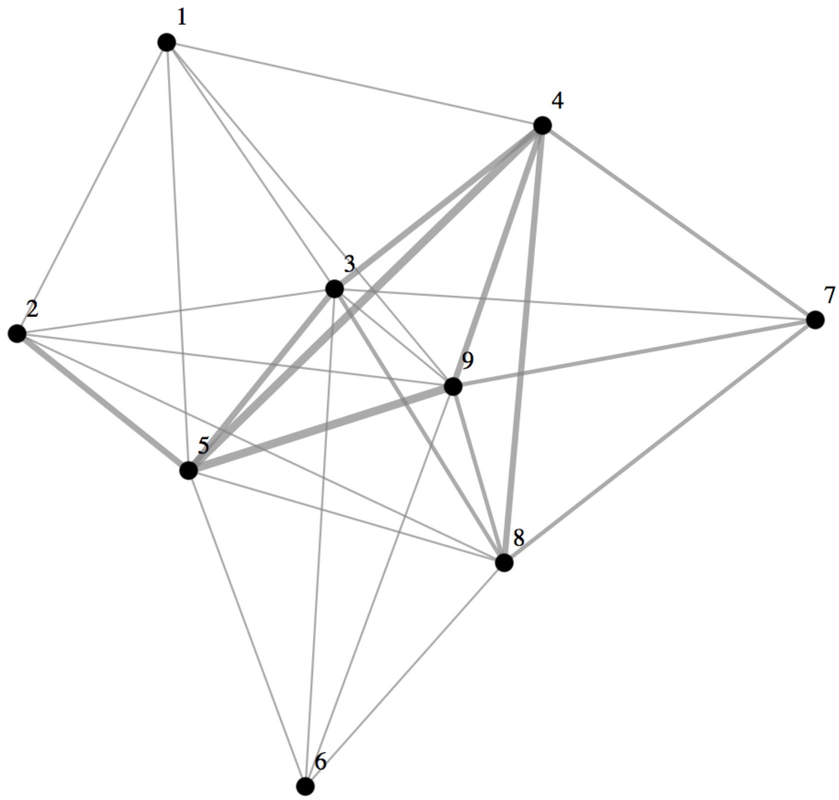


Figure 6.21: Network graph of a real-life wild capuchin monkey group.

Table 6.3 shows the values of edge count, vertex count and vertex degree for the social networks of real-life primate groups and our simulations. Results show that the networks created in the simulations have similar characteristics to the real-life primate networks also in terms of these parameters. The mean for the ratio of edges to vertices is 3.77 with values ranging from 3 to 4.08. The value of this ratio for our simulations is 3.83. Therefore the density of the simulated social network is similar to the densities of real-life primate networks. The vertex degree of our simulations is only 3% higher than the mean vertex degree, which shows the realistic social interaction structure of the simulated primate network.

Table 6.3: Network properties of analyzed ape groups

Parameter	Chimp-1	Chimp-2	Simulation results	Macaque-1	Macaque-2	Capuchin
Edge Count	49	46	49	37	33	27
Vertex Count	12	12	12	10	9	9
Vertex Degree	8.17	7.67	8.17	7.4	7.33	6

6.3.1.5 Subgroups in the network

Modularity-based clustering is applied in this set of simulations. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. Biological networks exhibit a high degree of modularity [180]. The modularity can be used for detecting community structure in social networks of primates.

Application of modularity-based clustering to the groups analyzed in simulation study showed that each network has either two or three subgroups. The number of individuals in these groups has a range between three and six. Fig. 6.22 shows two subgroups of the chimpanzee social network depicted in Fig. 6.17.

Fig. 6.23 shows three subgroups of a gorilla social network, the graph of which is presented in Fig. 6.18. There are subgroups in all primate social networks. For instance, a mother and her offspring would form a social subgroup. Essentially, the detection of these subgroups improves the investigation of the groups.

Fig. 6.22 and Fig. 6.23 are important for the detection and presentation of primate communities, which form small cohesive groups. Therefore, these small groups provide information on the community structure additional to the roles determined by our approach.

Subgroup information is also useful for the next installation or replacement of sensor and actor nodes in the same network. For instance, the number of actor nodes can be arranged according to the number of subgroups to improve data collection.

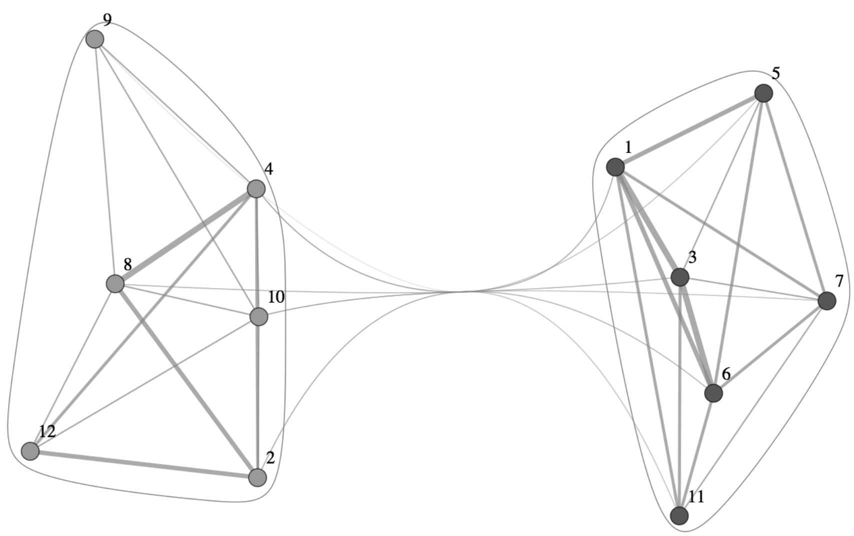


Figure 6.22: Subgroups of the chimpanzee network (1).

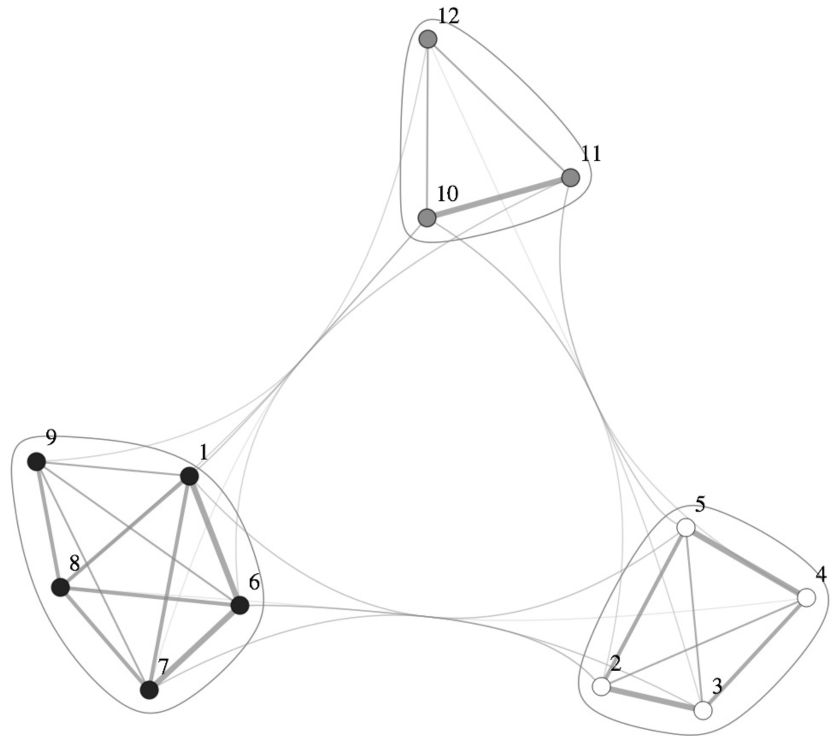


Figure 6.23: Subgroups of the network created by our simulations.

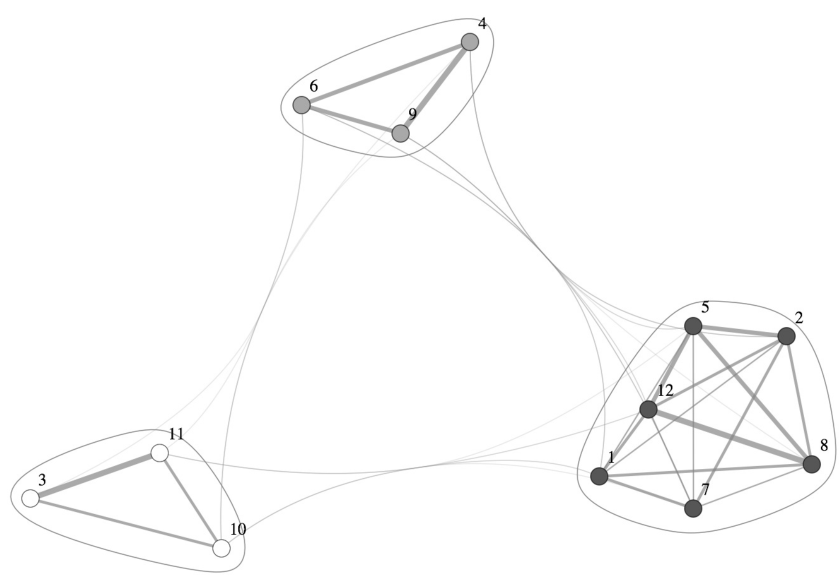


Figure 6.24: Subgroups of the chimpanzee network (2).

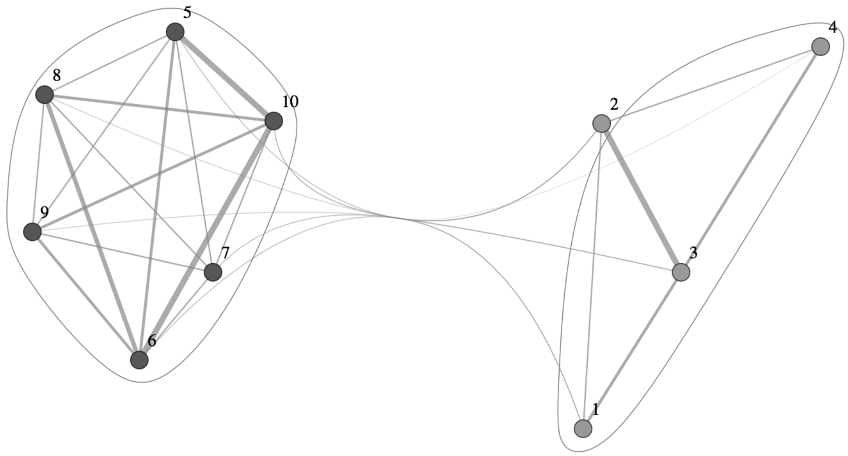


Figure 6.25: Subgroups of the macaque network.

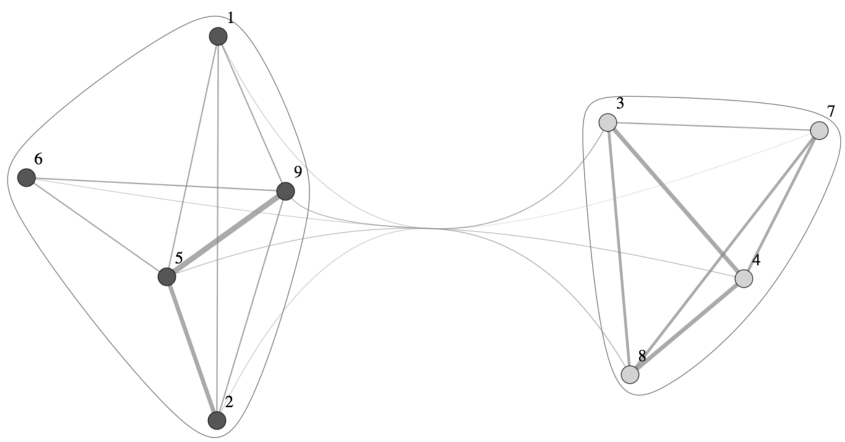


Figure 6.26: Subgroups of the capuchin monkey network.

6.3.2 Social network and friend ranking

In this section, we evaluate our social network and friend ranking approach by measuring how the patterns in the estimated friendship networks vary with the ranking algorithm and interaction characteristics. The Nodobo study includes call, SMS and proximity records.

The direction of the calls and messages are in the record along with the associated phone number and the duration of the call or length of the message. The participants of the study used provided mobile phones to communicate with their personal contacts as well as the other participants of the study.

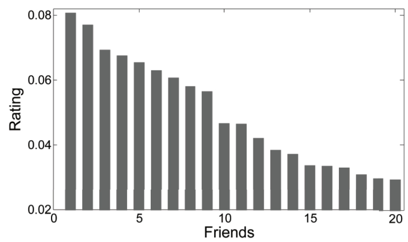
In our approach, we aim to generate the social network with all contacted friends for each participant. Therefore, the data must be analyzed and used for all friends of each user. Since the proximity data exist only for the current participants, it cannot be used in the same analysis with the call and SMS records. Additionally, there are false positive ties formed in the social graph when the proximities of the Nodobo dataset are used [181].

We use different weight values for calls and texts depending on the studies of Okdie et al. [175] and Boucher et al. [182]. According to these studies, users declare a higher satisfaction level in vocal communication compared to the communication based on texts. Therefore, we assigned a higher weight to the calls, $w_c = 1.25$, than SMS, $w_s = 1$, in our simulations. The data collection period is divided into intervals of one month and the interactions of the users are compared for each month to calculate the wins and losses of the friends. Table 6.4 summarizes the parameters used in the study.

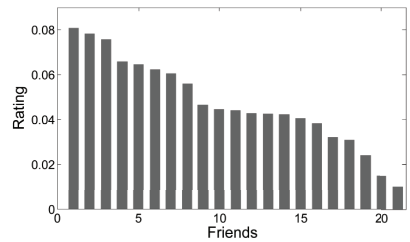
Table 6.4: Simulation parameters

Total time	4 months
Game period	1 month
Number of users	27
Call weight (w_c)	1.25
SMS weight (w_s)	1
Number of Calls	1309
Number of SMS	25,982

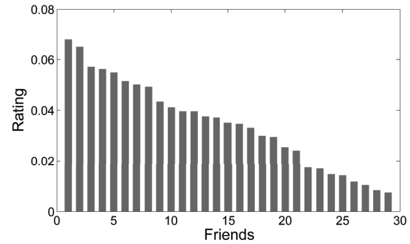
Fig. 6.27 shows the resulting ratings of friends with Colley method for nine of 27 users. The friends of the users are numbered and sorted according to their ratings in the figures. The pattern of the rating distribution reflects the discrete groups of friends for users. The resulting rating graphs have the same concavity characteristics, which is an effect of the nonsensitive Colley ranking method. This pattern also exists in the rating results of all students which are not included in Fig. 6.27. For 20 users ($\simeq 74\%$), friends can be grouped into three discrete groups such that friend with maximum rating in an intimacy group has a rating at least 15% lower than the friend with the minimum rating in the higher intimacy group.



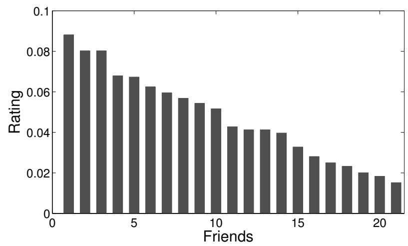
(a) User-1, Colley method



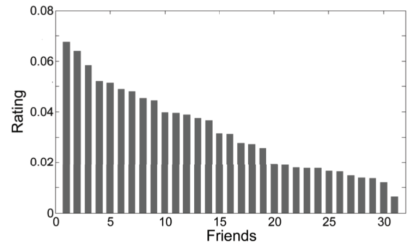
(b) User-2, Colley method



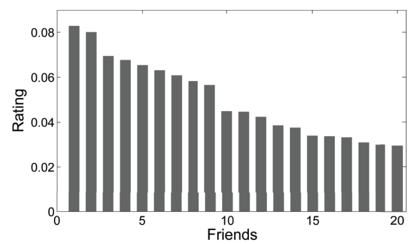
(c) User-3, Colley method



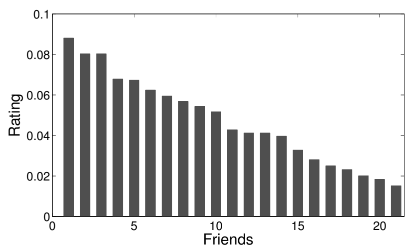
(d) User-4, Colley method



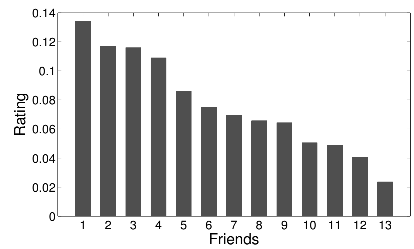
(e) User-5, Colley method



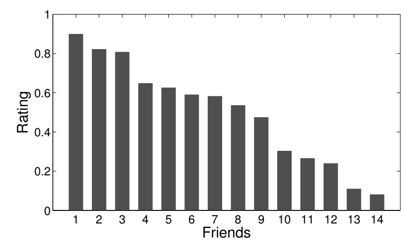
(f) User-6, Colley method



(g) User-7, Colley method



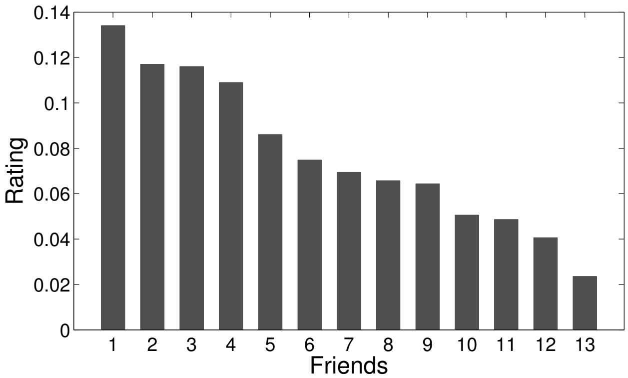
(h) User-8, Colley method



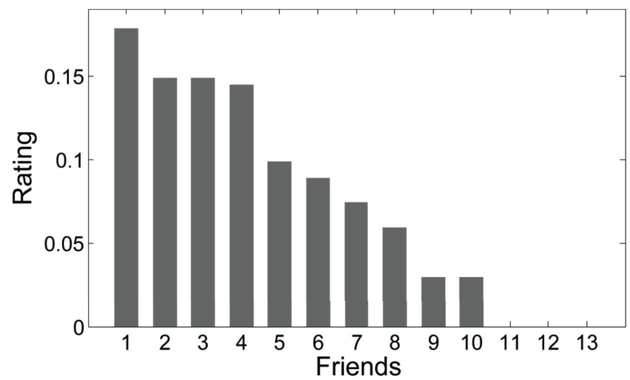
(i) User-9, Colley method

Figure 6.27: Ratings of friends with Colley method for nine users.

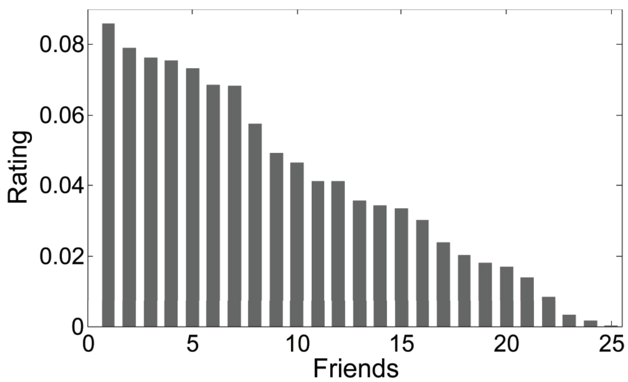
In the second set of the analyses, the ratings are calculated by using the Winning Percentage method. The experiment interval is divided into periods of one month. Fig. 6.28 shows a comparison of ratings by Colley and Winning Percentage method for two users. Both methods rank friends close to each other. However one of the main disadvantages of Winning Percentage method is that some of the users are not included in the friend list. Although the user communicated with these friends, they didn't have an interaction value considered as a win to be included in the friends list. Additionally, the pattern of discrete groups of friends cannot be observed in Winning Percentage method. The ratings of the friends are distributed in a more continuous pattern compared to the ratings of Colley method.



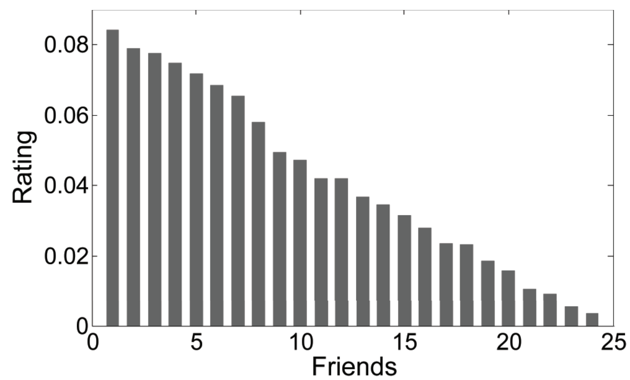
(a) User-1, Colley method



(b) User-1, Win. perc. method



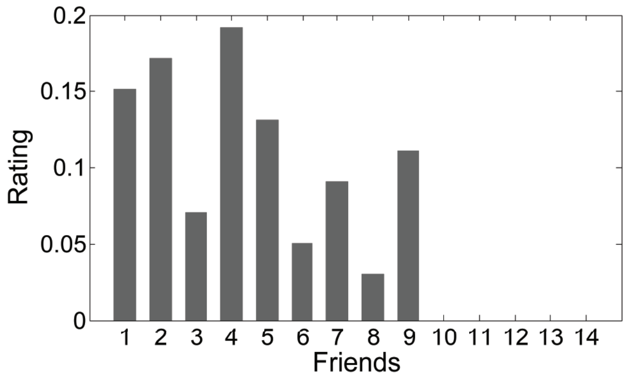
(c) User-2, Colley method



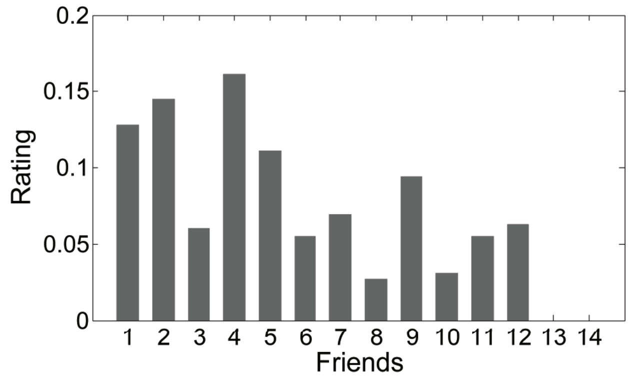
(d) User-2, Win. perc. method

Figure 6.28: Friend ratings for two users.

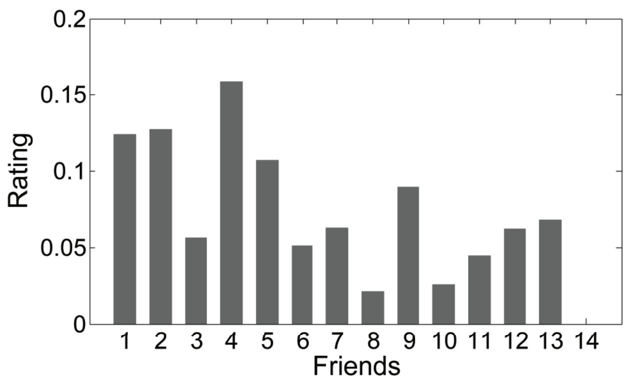
In Fig. 6.29, the change in the ratings of a user's friends is demonstrated for each month during the four month interval. Since the number of communicated contacts increases over time and the rankings change accordingly, the ratings are not sorted in Fig. 6.29 to keep the order of friends. The user communicates at most with newly added friends after each month, which can be observed in the results. However, the ratings do not change drastically and Colley method generates a consistent friend network through four months.



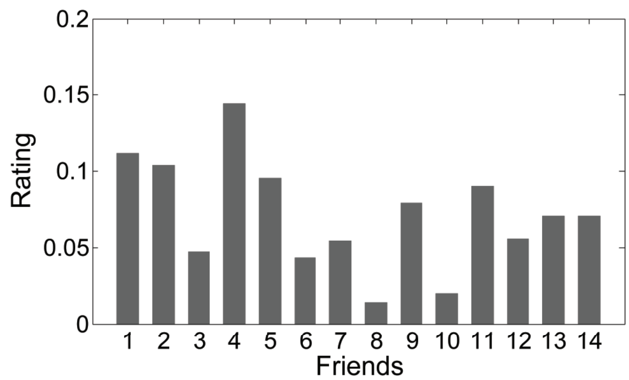
(a) Month 1



(b) Month 2



(c) Month 3



(d) Month 4

Figure 6.29: The change in friend ratings in four months.

6.3.3 Trust based social network

We use Reality Mining Database [183] to discuss the application of trust based ranking on real data. Reality Mining is a mobile phone database, which represents over 350000 hours of continuous data collected from 100 students for nine months. The information collected in Reality Mining includes call logs, SMS messages, Bluetooth proximity information, cell tower IDs, application usage, and phone status. Data also include durations and directions of calls and text messages. Hence the database incorporates detailed information on communication among users in forms of text and voice. Data from Bluetooth devices and cell towers enable approximation of face-to-face communication time and duration information. The number of contacts is more than 50 for the majority of users in the study, which creates an appropriately sized ego network for each person to analyze. The analysis is done using text messages, phone calls and Bluetooth proximities as different types of interactions.

To comply with SBH model, a small portion of friends must have strong ties, a larger number of friends must have medium ties and most of the friends must have weak ties with the ego. Hence the distribution of relationship strength for each person must present a power-law curve. As an initial test, we assigned higher weights to voice calls and proximities when they last longer than average interaction durations. The analysis results show that even with these conditions, we observe power-law distribution of interaction (friendship) values for each ego network in the database. The overall distribution of interactions for all users in the network is also power-law as shown in Fig. 6.30.

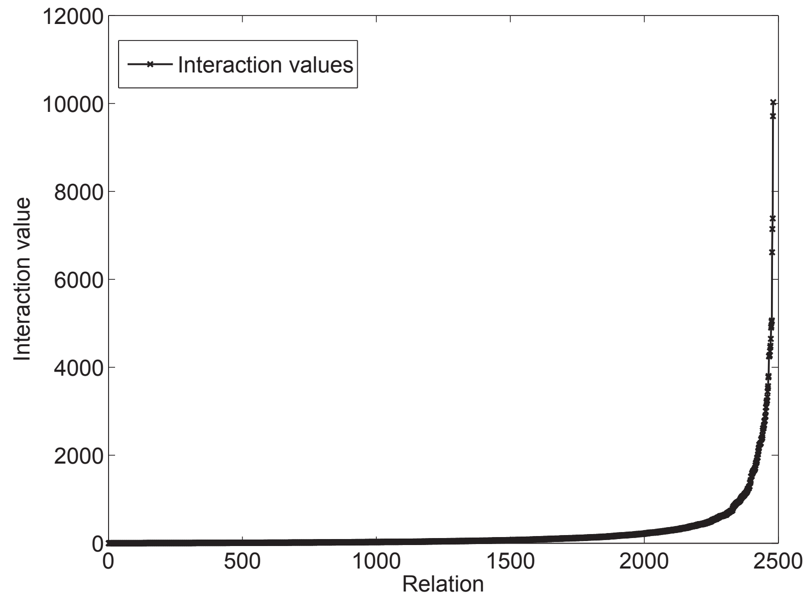
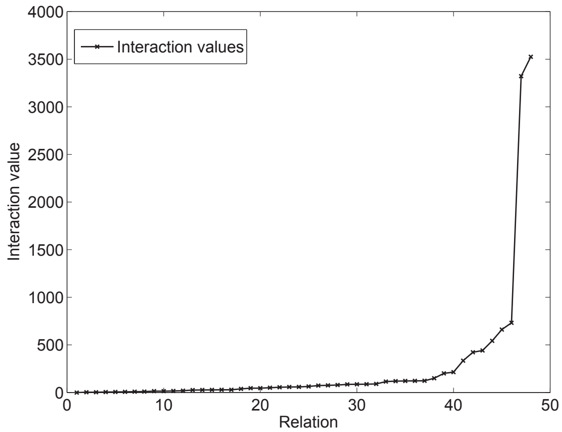
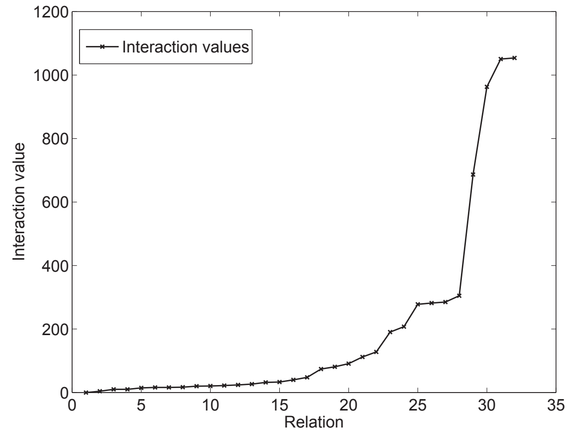


Figure 6.30: Distribution of interaction values for the whole network.

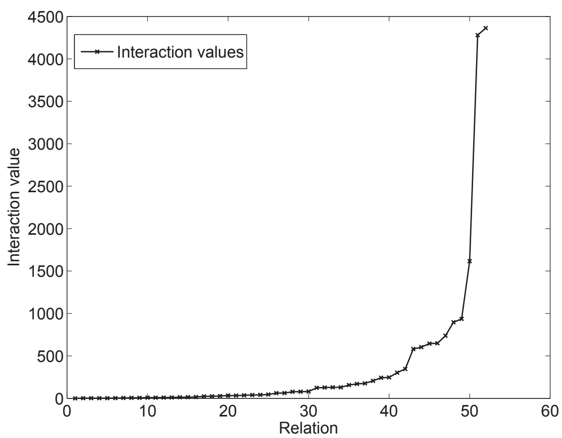
In another simulation set, the weight values of the interactions are taken to be equal and the interaction values calculated for four randomly selected users. Fig. 6.30 shows the relations of these fours users and the interaction values for their relations. We observe a similar distribution of interaction values for each individual. Another observation is that the maximum interaction value changes from one person to another while the number of strong ties stay in the range from one to three for all of the students. We also observed that the number of very close friends is usually 2 or 3. For accurate social network interpretation and generation of the circles, the range of all possible weight values must be determined. When the voice and proximity are prioritized over text communication by taking the duration of the interaction into account, the distinct friend groups of intimacy levels with power-law distribution would become more apparent.



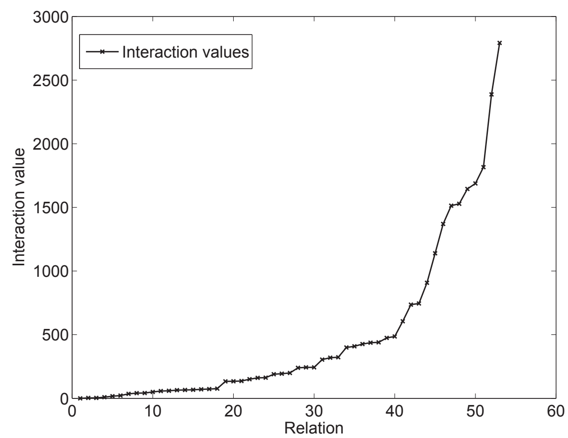
(a) User 1



(b) User 2



(c) User 3



(d) User 4

Figure 6.31: The interaction values of the relations for four users.

CHAPTER 7
ACTOR POSITIONING IN AERIAL SENSOR NETWORKS

In this chapter we present APAWSAN, an actor positioning strategy for aerial sensor and actor networks (AWSANs) [23–26]. In AWSANs, UAVs have been equipped with additional sensors for collecting environmental data. Therefore, UAVs have become popular in a wide range of applications which benefit the environmental monitoring, air formation measurement, search and rescue and so on. With the development of aerial networks with multiple UAVs, some of these applications such as toxic plume observation or atmospheric sensing of storm dynamics focus on continuous investigation of three dimensional (3D) space.

Although current approaches mostly use UAVs in solo flight, there are emerging concepts for employing multi-UAV systems. Compared to single-UAV systems, multi-UAV systems have several advantages such as scalability and survivability. Multi-UAV systems facilitating the communication among UAVs are more flexible and can extend the coverage of the system. These systems either act autonomously or depend on communication with an infrastructure such as a satellite base or a ground station for the operation of the UAVs. There are several approaches for multi-UAV systems with heterogeneous node structures ([184], [185], [186]).

In AWSANs, UAVs acting as sensor nodes are generally smaller and they only collect data from the environment. Some of these UAVs have the size of a fulmar and they are inexpensive compared to fully equipped research aircrafts, which act as actor nodes. In addition to data collection, actor nodes also act on the environment by using actuators such as servo-mechanisms. For instance, low-flying helicopter platform by Thrun et al. [187] provides ground mapping and air-to-ground cooperation of autonomous robotic vehicles. Besides acting on the environment and collecting data, actors perform networking functionalities such as processing or relaying of data in multi-UAV solutions and they acquire a hierarchically higher role in the network for these applications.

The formation and dynamic adaptation of the network topology in 3D space is important for the coverage of 3D environment and the effective data collection. Sensor networks have been utilized recently for applications in 3D space such as space exploration [185], airborne [140] and underwater surveillance [131]. However, these solutions in different domains do not directly apply to UAV systems, which have characteristic constraints. For instance, the dynamic environmental conditions, node movements and terrain structures complicate the maintenance of communication links. The nodes in AWSANs are mobile with higher speeds compared to most other systems. Therefore, the topology changes are also more frequent.

The radio links and physical layer characteristics are also different in AWSANs since the communication ranges are generally longer than underwater sensor networks or vehicular networks. The communication range of each UAV is determined by the range of wireless radio it is equipped with. Hence the network topology design affects the 3D coverage of the network. The signal strengths of antennas depend on their orientations. Actor UAVs can not have the same signal strength at every position even if they are dynamically positioned in the network. Therefore, a positioning strategy for aerial WSANs must be able to attain and maintain 3D coverage of the observed environment while taking these constraints into account.

The considered scenario in this chapter is a volcanic eruption such as the eruptions of the volcano Eyjafjallajökull at Eyjafjöll in Iceland in the Spring of 2010. The erupting lava of Eyjafjallajökull injected a cloud of ash into the Jet Stream. Even when the ashcloud was carried away from the volcano with the wind, it was not possible to bring personnel for close-up observation of the volcano or the plume. The UAV system we present here consists of a central UAV acting as the sink, small UAVs with sensor nodes and larger actor UAVs collecting the data. This system can form a unique three dimensional system for environmental monitoring and can be used for high quality observation of toxic plume behavior.

APAWSAN is presented to achieve effective 3D volume coverage while preserving 1-hop connectivity from each actor UAV to a central sink UAV. Our positioning algorithm is based on the Valence Shell Electron Pair Repulsion (VSEPR) theory [155]. VSEPR theory has been used in chemistry to describe the atom alignments in a molecule around a central atom and to develop molecular mechanics force fields [188]. We utilize VSEPR theory to define the positions of actors with respect to the sink. Then we extend our approach for multiple sinks using the VSEPR theory principles for multiple central nodes and the network among these central nodes. The positioning strategy is adapted to the characteristics of UAV systems and aforementioned constraints by using a rotatable hybrid antenna model (O-BESPAR), which is chosen according to the real life experiments conducted for the possible antenna modules [189]. The requirements of our approach such as efficient neighbor discovery and non-interference communication are achieved by the (O-BESPAR) [189] and beamforming.

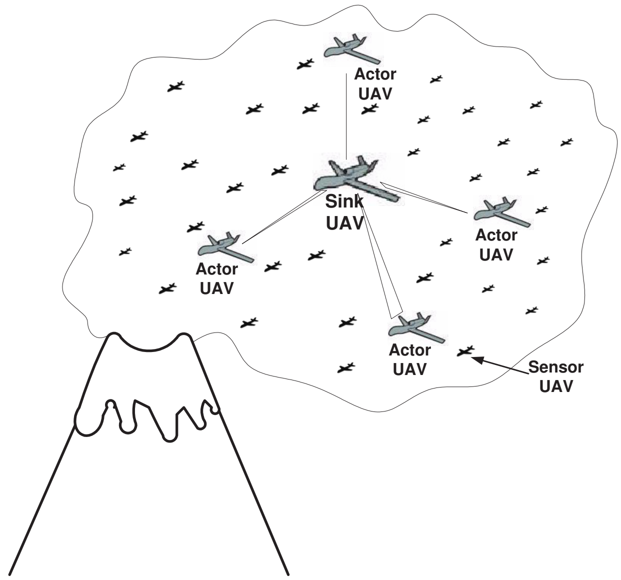


Figure 7.1: Volcanic plume application scenario.

7.1 System model

The system consists of N nodes, with a set, S, of small UAVs, which have built-in sensor nodes and a set, A, of more powerful UAVs with actor nodes. There is also a sink node which is located on the largest sized UAV with the extended capabilities so that it is not affected by the expected or unprecedented environmental conditions.

The formation of actor-sink backbone and affiliation of sensor nodes with the actors are similar to SOFROP [14] and nodes do not follow any predetermined initial configuration. Each sensor node s communicates only with direct neighbors $Neigh(s)$ and keeps a "$weight''$ value, which is "$k -$ (hop value)$''$ where k is the *weight* of the actor and hop value is the number of hops required to reach the affiliated actor. The sensor nodes and actors are assumed to have transmission radii r_s and r_a, respectively, with spherical transmission ranges.

7.2 Actor positioning

7.2.1 VSEPR theory approach

Our positioning strategy preserves 1-hop connectivity between each actor and the sink by using the VSEPR theory of chemistry. The VSEPR model is the most successful model for the prediction of closed-shell molecule geometries. Laplacian of the charge density provides the physical basis for the VSEPR model. VSEPR model is used for actor positioning in an aerial wireless sensor and actor network.

According to VSEPR theory, the Laplacian of the electronic charge density exhibits extrema in the valence shell of the central atom within a molecule. These extrema indicate the presence of localized concentrations of electronic charge. The spherical surface on which the electron pairs are assumed to be localized is identified with the sphere of maximum charge concentration in the valence-shell charge concentration and the localized pairs of electrons are identified with the local maxima.

VSEPR theory uses the "AXE method" of electron counting, in which A is the number of central atoms, X is the number of sigma bonds between the central atom and the surrounding atoms and E is the number of lone electron pairs. The geometry predictions depend on the steric number, which is the sum of X and E. E is used particularly for deciding the positions of the actors in systems with multiple sinks in our approach.

VSEPR theory is applied such that the possible actor positions for different number of actors are determined according to VSEPR theory geometries, which are given in Fig. 7.2. Then these locations are converted into positions with respect to the sink. The number of actors (n) and the communication ranges of the nodes are the most important factors defining the locations of actors.

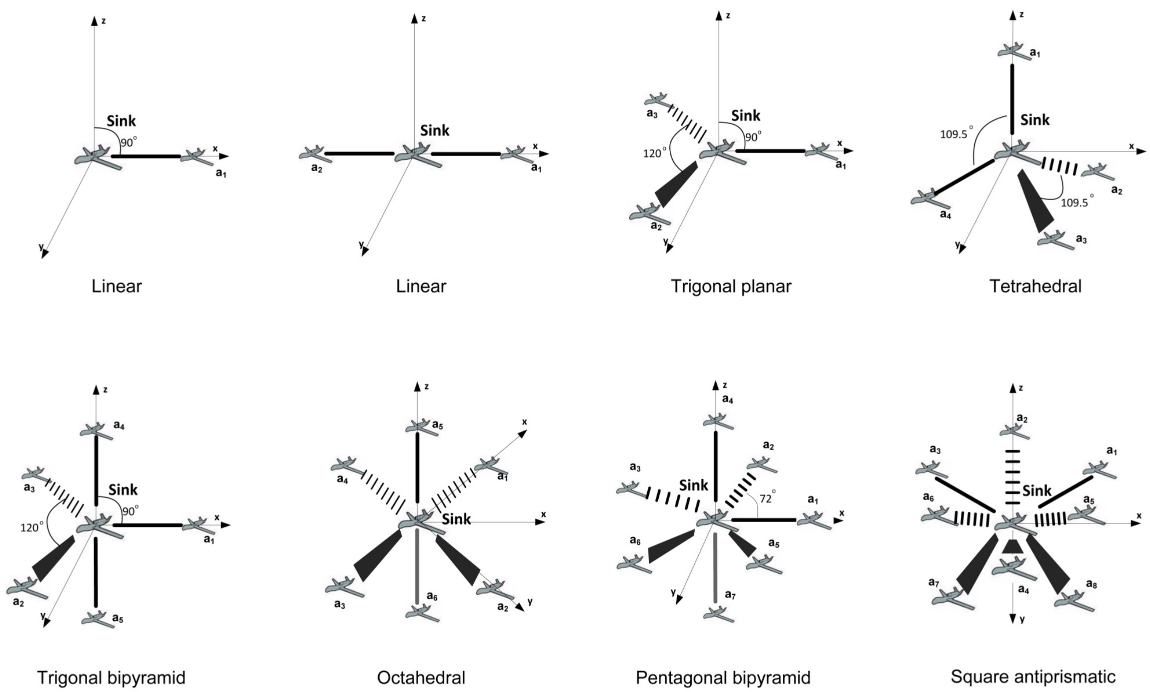


Figure 7.2: VSEPR theory geometries.

7.2.1.1 VSEPR theory geometries of WSAN

The geometries formed by the UAVs using the VSEPR theory are identified by creating a coordinate system and taking the position of the sink, p_S, as the origin of this coordinate system. The main direction in the flight plan of sink forms the x-coordinate and the positions of the actors during the flight are defined with respect to the sink and its flight direction. The formulation of geometries is important for the definition of positions that the actors can be located and for the definition of transitions between geometries. The locations of the actors according to the defined coordinate system. The peripheral atoms in VSEPR theory are mapped to the actors and the central atom is mapped to the sink. The possible actor positions for different number of actors are found and these positions are converted into positions with respect to the sink in order to be used during flight. A designed algorithm allows the actors to position during the flight and for the transitions from one geometry to the other. Hence both for increasing and decreasing number of actors, there is a self-organizing system of UAVs handling sudden failures or temporary communication loss.

When there is a single actor, it takes a position with a predefined distance of r to the sink. Similarly in the case with two actors, the sink and actors are arranged during the flight with an expected connection angle of 180°. This geometrical arrangement is called "Linear" geometry (Fig. 7.3). The positions of the actors in Linear geometry are as follows:

$$p_{a_1}(x, y, z) = (r, 0, 0) \quad p_{a_2}(x, y, z) = (-r, 0, 0) \tag{7.1}$$

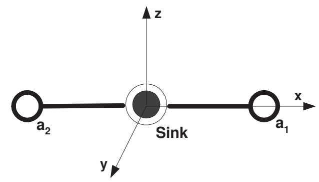


Figure 7.3: VSEPR theory linear geometry.

The flight model when there are three actors around the sink is determined by the molecular geometry model with a single atom at the center and three peripheral atoms at the corners of a triangle all in one plane. This is called as “Trigonal planar” (Fig. 7.4) and its characteristic property is the connection angles of 120° between two actors. These actors are at identical distances to the sink with positions given as follows:

$$
\begin{aligned}
&p_{a_1}(x, y, z) = (r, 0, 0),\\
&p_{a_2}(x, y, z) = (-r.sin(30^\circ), r.sin(60^\circ), 0)\\
&p_{a_3}(x, y, z) = (-r.sin(30^\circ), -r.sin(60^\circ), 0)
\end{aligned}
\tag{7.2}
$$

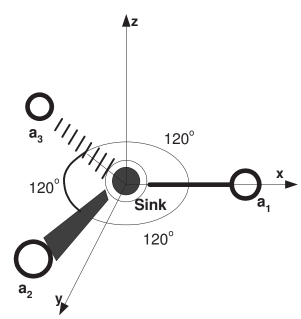


Figure 7.4: VSEPR theory trigonal planar geometry.

When there are four peripheral actor UAVs, the sink is located at the center with four substituents that are located at the corners of a tetrahedron. This geometry is called "Tetrahedral" (Fig. 7.5) and the connection angles are $cos^{-1}(-1/3) \approx 109.5°$ when all four actors are located at the positions calculated according to tetrahedral geometry.

$$
\begin{aligned}
&p_{a_1}(x,y,z) = (0,0,r),\ \ p_{a_2}(x,y,z) = (-r.a, -r.b, r.cos(109.5°)) \\
&p_{a_3}(x,y,z) = (-r.sin(109.5°), 0, r.cos(109.5°) \\
&p_{a_4}(x,y,z) = (-r.a, r.b, r.cos(109.5°))
\end{aligned}
\tag{7.3}
$$

where $a = sin(109.5°).sin(30°)$, $b = sin(109.5°).cos(30°)$

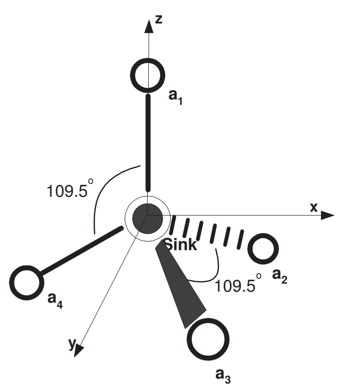


Figure 7.5: VSEPR theory tetrahedron geometry.

When there are five actors surrounding the sink, they take positions during the flight with non-identical connection angles relative to the sink. There is no geometrical arrangement for one sink and five actors, which results in five equally sized connection angles in three dimensions. In trigonal bipyramid geometry, three actors are positioned on the $y = 0$ plane with connection angles of 120° whereas the other two actors take positions on y-axis with angles of 90° to the $y = 0$ plane. Hence the geometry formed with each actor at a corner of this geometry is a "Trigonal bipyramid" (Fig. 7.6).

$$
\begin{aligned}
&p_{a_1}(x,y,z) = (r,0,0),\ \ p_{a_2}(x,y,z) = (-r.sin(30^\circ), r.sin(60^\circ), 0)\\
&p_{a_4}(x,y,z) = (0,0,r),\ \ p_{a_3}(x,y,z) = (-r.sin(30^\circ), -r.sin(60^\circ), 0)\\
&p_{a_5}(x,y,z) = (0,0,-r)
\end{aligned}
\tag{7.4}
$$

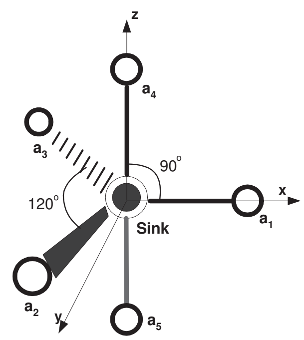


Figure 7.6: VSEPR theory tetrahedron geometry.

When there are six actors, they are arranged around the sink symmetrically, defining the vertices of an octahedron as given in Fig. 7.7. The octahedron has eight faces as its prefix implies and the final geometry is an Octahedral with an actor at each corner.

$$
\begin{aligned}
&p_{a_1}(x,y,z) = (r,0,0) \quad p_{a_2}(x,y,z) = (0,r,0)\\
&p_{a_3}(x,y,z) = (r,0,0) \quad p_{a_4}(x,y,z) = (0,-r,0)\\
&p_{a_5}(x,y,z) = (0,0,r) \quad p_{a_6}(x,y,z) = (0,0,-r)
\end{aligned}
\tag{7.5}
$$

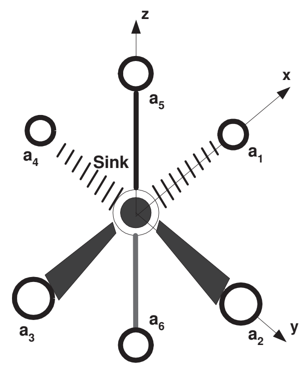


Figure 7.7: VSEPR theory octahedron geometry.

There are several possible geometries when seven actors exist in the network such as the mono-capped octahedron, mono-capped trigonal prism and the pentagonal bipyramid (or dipyramid). "Pentagonal bipyramid" (Fig. 7.8) is the chosen geometry since its suitable for transition between geometries in cases such as the loss or an addition of an actor. Pentagonal bipyramid defines the molecular geometry with one atom at the center with seven ligands at the corners of a pentagonal dipyramid. The connection angles are not identical for actors in this geometry.

$$
\begin{aligned}
&p_{a_1}(x,y,z) = (r,0,0), p_{a_2}(x,y,z) = (r.cos72^\circ, r.sin72^\circ, 0)\\
&p_{a_4}(x,y,z) = (0,0,r), p_{a_3}(x,y,z) = (-r.cos36^\circ, r.sin36^\circ, 0)\\
&p_{a_7}(x,y,z) = (0,0,-r), p_{a_5}(x,y,z) = (r.cos72^\circ, -r.sin72^\circ, 0)\\
&p_{a_6}(x,y,z) = (-r.cos36^\circ, -r.sin36^\circ, 0)
\end{aligned}
\tag{7.6}
$$

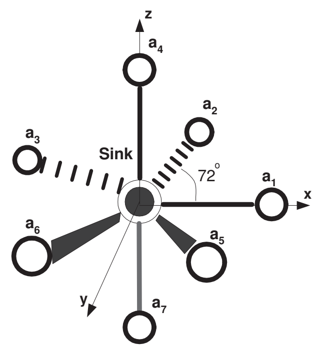


Figure 7.8: VSEPR theory pentagonal bipyramid geometry.

There are examples in VSEPR theory with eight surrounding nodes, which maximize the distance to the nearest point, or use electrons to maximize the sum of all reciprocals of squares of distances. According to the VSEPR theory, the square antiprism (See Fig. 7.9) is the favored geometry among the possible geometries with eight surrounding atoms. A square anti-prism corresponds to the shape when eight points are distributed on the surface of a sphere with the aim of maximizing the distance between each pair.

$$\begin{array}{ll}
p_{a_1}(x,y,z) = (r.a\frac{\sqrt{2}}{2}, 0, r.\frac{h}{2}) & p_{a_2}(x,y,z) = (0, r.a\frac{\sqrt{2}}{2}, r.\frac{h}{2}) \\
p_{a_3}(x,y,z) = (-r.a\frac{\sqrt{2}}{2}, 0, r.\frac{h}{2}) & p_{a_4}(x,y,z) = (0, -r.a\frac{\sqrt{2}}{2}, r.\frac{h}{2}) \\
p_{a_5}(x,y,z) = (r.a, r.a, -r.\frac{h}{2}) & p_{a_6}(x,y,z) = (-r.a, r.a, -r.\frac{h}{2}) \\
p_{a_7}(x,y,z) = (-r.a, -r.a, -r.\frac{h}{2}) & p_{a_8}(x,y,z) = (r.a, -r.a, -r.\frac{h}{2})
\end{array} \tag{7.7}$$

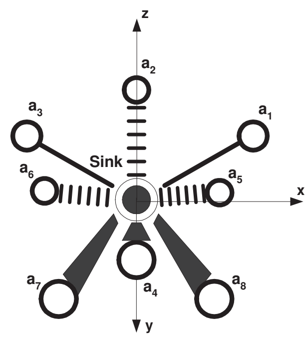


Figure 7.9: VSEPR theory square antiprism geometry.

where a and h are constants used in pentagonal bipyramid geometry to simplify the transitions. $h/2 \approx 0.5237$ represents the positive and negative z values for the planes that the actors are located at and $a \approx 1.2156$.

The positioning algorithm for actors, which is designed according to VSEPR theory principles and these calculated positions, is given in Algorithm 8. The positioning calculations assume spherical transmission and reception ranges with identical RSSI and loss rates at every communication angle. However, these factors are very important and effective in the performance of real life UAV systems. The real-life challenges of UAV systems must be taken into consideration when designing a positioning model. Therefore the characteristics of the antennas used on the actors and the sink are critical. We improved the positioning method by extending it with a realistic antenna model and by modifying the positioning algorithm accordingly.

Algorithm 8 Actor positioning

r: Distance from an actor to the sink, $\vec{p}$: Position vector of a node,
Θ: Angle between $\vec{p}_s$ and $\vec{p}_i$, $(x_{sink}, y_{sink}, z_{sink})$: Coordinates of the sink
if $n < 4$ **then**
  One actor is positioned at $(x_{sink} + r, y_{sink}, z_{sink})$
  $i = \Theta$
  while $i < 360°$ **do**
    Next actor is positioned at $(x_{sink} + r.cos(\Theta), y_{sink}, z_{sink})$
    $i = i + \Theta$
  end while
else if $n = 4$ **then**
  One actor is positioned at $(x_{sink}, y_{sink}, z_{sink} + r)$
  $i = 0°$, $\Phi = -19.471°$
  while $i < 360°$ **do**
    $x_a = r.cos(i).cos(\Phi) + x_{sink}, y_a = r.sin(i).cos(\Phi) + y_{sink}$
    $z_a = r.sin(\Phi) + z_{sink}$
    $i = i + 120°$
  end while
else if $8 > n > 4$ **then**
  Two actors are positioned on (x_{sink}, y_{sink}) line
  One of remaining actors is positioned at $(x_{sink} + r, y_{sink}, z_{sink})$
  $\Theta = \frac{360}{n-2}$ on $z = 0$ plane, $i = \Theta$
  while $i < 360°$ **do**
    Next actor is positioned at $(x_{sink} + r.cos(i), y_{sink} + r.sini, z_{sink})$
    $i = i + \Theta$
  end while
else if $n = 8$ **then**
  $\frac{h}{2} = 0.5237$, $a = 1.2156$:
  $x_{sink} \pm r.a, y_{sink} \pm r.a, z_{sink} - r.\frac{h}{2}$ for first four actors
  $x_{sink} \pm r.\frac{a}{\sqrt{2}}, y_{sink}, z_{sink} + r.\frac{h}{2}$ for two actors
  $x_{sink}, y_{sink} \pm r.\frac{a}{\sqrt{2}}, z_{sink} + r.\frac{h}{2}$ for remaining actors
end if

7.2.2 O-BESPAR antenna model

Omni Bi-directional ESPAR (O-BESPAR) antenna model, which leverages the complementary properties of omni-directional and directional antennas. While omni-directional antennas enable 360 degrees of coverage when needed, the directional antennas provide high throughput and low interference. In addition to this important unifying characteristic, O-BESPAR is utilized for our flight model based on its several other properties, which can be summarized in three main parts as follows:

- Utilization of two independent directional beams permit a node to transmit and receive simultaneously.
- The lightweight and small size of the module make it rotatable so that the beamforming can be steered to any direction in 3D.
- Cooperation of omni-directional and beamforming antennas permits capability changes for data transferring.

Our communication protocol incorporates an efficient neighbor discovery mechanism, which not only allows UAVs to discover each other rapidly but also enables quick alignment of directional beams to maximize the data transfer opportunities. According to our communication protocol, sender UAV broadcasts control messages through the omni module in order to exchange location information with receiver. After both beams are steered to each other, the data transmission commences over the directional module. However, the transmission range of the omni antenna is much smaller than the directional ESPAR module. Therefore if no neighbor is found by the broadcast of the omni module, the communication protocol uses the directional module to perform bi-directional beam sweeping. Each beam covers 180 degrees so that the scanning delay is minimized.

GPS receiver and altitude sensor are commonly used as built-on equipments for UAVs nowadays. After using omni-directional antenna module to locate the sender UAV, sink UAV has to calculate the angle difference to steer one of its directional beams to the sink. The angle, $(\phi, \theta)_{A,S}$, between the sink, S, and an actor, A, is calculated as follows:

$$(\phi, \theta)_{A,S} = \arctan\Big(\frac{z_A - z_S}{\sqrt{(x_A - x_S)^2 + (y_A - y_S)^2}}\Big)$$

ϕ and θ stand for the angle of beamforming in horizontal and vertical plane separately.

We utilize two-ray ground path loss as the propagation model. According to this model, the receiving power P_r depends on the transmission power P_t, antenna gain of transmitter G_t, antenna gain of receiver G_r, distance between the actor and sink $d_{A,S}$, the wavelength λ and the antenna positions H_t and H_r. The calculation of the receiving power P_r is given as follows:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2}[4\sin(\frac{\pi H_t H_r}{\lambda d})]^2$$

The effect of omni antenna orientation on the RSSI value is demonstrated with experiments [190]. Thus, the omnidirectional antenna is not completely isotropic in a 3D network.

In particular, the measurement of RSSI is extremely important for the positioning schemes for two reasons. First, the neighbor discovery packets are sent from the omni antenna module. If the RSSI value is too low, actor UAV has to sacrifice the coverage to fly closer to the sink. Second, the actor UAVs fly around the sink and keep their positions in the flight. If the sink UAV does not receive the beacon messages from some actor for time out, that actor UAV is assumed to be lost and the network topology is changed to a non-optimal geometry.

7.2.3 Communication and rearrangement protocols

We propose two protocols for the communication between the sink UAV and actor UAVs. Particularly, there are three main objectives of these protocols. First, the control packets (Beacon and ACK) and data packets are transmitted by the omni and directional module separately. Thus, omni module performs fast neighbor discovery and directional module guarantees successful data delivery. Second, the protocol works with our actor positioning algorithm to form and change the VSEPR geometries. In other words, the sink UAV achieves dynamic positioning for the actor nodes. Third, during the flight, the sink UAV must respond to the change of VSEPR geometry efficiently. Since some actors may leave or join the network, the protocol includes a repositioning mechanism which rearranges the actors and updates the beamforming direction.

The main algorithm for the communication between actor and sink UAVs is presented in Algorithm 9. Before transmitting the collected data to the sink, an actor node uses omni antenna module to broadcast beacon messages, which contain its identification (id) and current 3D location, (x,y,z) coordinates. When the sink receives the beacon messages, it checks the actor‘s connection record. If it is not found, the sink updates the number of neighbor actors and decides on the VSEPR geometry to be used by using Algorithm 10. Then the actors start changing their positions relative to the sink and create a new beamforming.

The communication links of omni module may fail occasionally due to various reasons such as obstacles, interference or low RSSI value caused by its orientation. Therefore, we also propose Algorithm 10 to make O-BESPAR antenna model adaptable to the dynamics of geometrical flight model. If the sink does not receive any beacon from an actor, which has been connected for $T_{timeout}$, the sink assumes that the actor has left the geometry. The sink sends out Position Update Message (PUM) to rearrange actors' positions with the purpose of maximizing network coverage.

Algorithm 9 Actor-Sink UAVs Communication Protocol

U_{actor}: actor UAV, U_{sink}: sink UAV, n: number of actor UAVs, $(\phi, \theta)_{A,S}$: the angle between actor and sink UAVs
if U_{actor} has data to transmit **then**
  Broadcast *Beacon* through omni module
  if U_{sink} received *Beacon* through omni module **then**
    Extract UAVId
    if UAVId had been recorded by U_{sink} **then**
      Reply ACK through omni module and direct one beam to U_{actor}
    else
      Update n, operate *Algorithm 7*, Calculate $(\phi, \theta)_{A,S}$
      Reply ACK through omni module and direct one beam to U_{actor}
    end if
    if ACK is received by omni module of U_{actor} **then**
      Calculate $(\phi, \theta)_{A,S}$, direct one beam to U_{sink}, Start transmitting data through directional module
    end if
  else if U_{actor} does not receive ACK **then**
    U_{actor} sends out *Beacon* through directional module
    if U_{sink} receives *Beacon* through directional module **then**
      Update n, operate *Algorithm 7*, Reply ACK message through omni module and direct one beam to U_{actor}
    end if
    if ACK is received by U_{actor} **then**
      Start transmitting data through directional module
    else
      Sweep the beam to another direction
    end if
    if No ACK is received by both antenna modules **then**
      U_{actor} stores the data into a buffer
    end if
  else if U_{sink} does not receive *Beacon* from U_{actor} for $T_{timeout}$ **then**
    U_{actor} is lost, Sink UAV runs *Algorithm 9* to update flight geometry
  end if
end if

Algorithm 10 UAVs Geometrical Rearrangement

U_{sink} broadcasts *Position Update Message (PUM)* through omni module

U_{sink} calculates $(\phi_{pum}, \theta_{pum})_{UAV}$ and directs one beam to the new position

if U_{actor} receives PUM **then**

Change the position and calculate $(\phi_{pum}, \theta_{pum})_{UAV}$

Direct one beam to the U_{sink}

end if

7.2.4 Extension of VSEPR theory approach

We extended our initial approach by using an approach from molecular geometry. The compounds in nature have less than eight peripheral atoms. Therefore the initial VSEPR theory was presented for one central atom and at most eight surrounding atoms. Most of the current applications of UAV systems are composed of less number of actor UAVs [184]. Therefore APAWSAN [23] employed only the basic VSEPR theory for actor positioning. However the number of nodes in the network and the total covered volume can be increased if this approach is improved by including more actor nodes. Gillespie [191] applied the rules of VSEPR theory for up to twelve actors and presented an application of the theory for these higher number of surrounding nodes around a central node. These geometries are formed depending on the same repulsion force rules used in initial VSEPR theory. Hence we extended our approach by utilizing VSEPR theory principles to allow deployment of more than eight actors.

The geometries for nine to twelve actors, shown in Fig. 7.10, are monocapped square antiprism, bicapped square antiprism, icosahedron minus one apex and icosahedron. The monocapped square anti-prism in our approach corresponds to the geometry with the one more actor location $p_{a_9}(x, y, z) = (0, 0, r)$ additional to square anti-prism. For bicapped square anti-prism, there is one more actor positioned at $p_{a_9}(x, y, z) = (0, 0, r)$. The icosahedron is a geometrical shape composed of twenty triangular faces, thirty edges and twelve vertices. Icosahedron minus one apex is an icosahedron with one missing vertex. The favored geometry for twelve actor geometry is a regular icosahedron with identical equilateral faces and following actor positions:

$$
\begin{array}{ll}
p_{a_1}(x, y, z) = (0, \frac{r}{2}, r.\frac{\sqrt{5}-1}{4}) & p_{a_2}(x, y, z) = (0, -\frac{r}{2}, r.\frac{\sqrt{5}-1}{4}) \\
p_{a_3}(x, y, z) = (0, \frac{r}{2}, -r.\frac{\sqrt{5}-1}{4}) & p_{a_4}(x, y, z) = (0, -\frac{r}{2}, -r.\frac{\sqrt{5}-1}{4}) \\
p_{a_5}(x, y, z) = (\frac{r}{2}, r.\frac{\sqrt{5}-1}{4}, 0) & p_{a_6}(x, y, z) = (-\frac{r}{2}, r.\frac{\sqrt{5}-1}{4}, 0) \\
p_{a_7}(x, y, z) = (\frac{r}{2}, -r.\frac{\sqrt{5}-1}{4}, 0) & p_{a_8}(x, y, z) = (-\frac{r}{2}, -r.\frac{\sqrt{5}-1}{4}, 0) \\
p_{a_9}(x, y, z) = (r.\frac{\sqrt{5}-1}{4}, 0, \frac{r}{2}) & p_{a_{10}}(x, y, z) = (-r.\frac{\sqrt{5}-1}{4}, 0, \frac{r}{2}) \\
p_{a_{11}}(x, y, z) = (r.\frac{\sqrt{5}-1}{4}, 0, -\frac{r}{2}) & p_{a_{12}}(x, y, z) = (-r.\frac{\sqrt{5}-1}{4}, 0, -\frac{r}{2})
\end{array}
$$

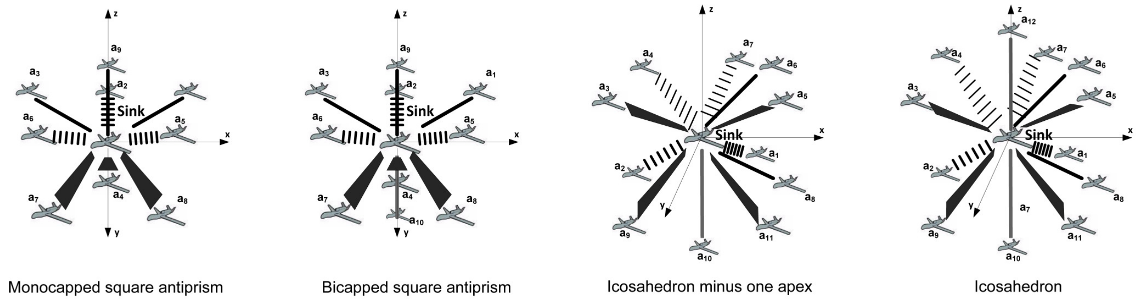


Figure 7.10: Geometries of nine to twelve actors.

The positioning algorithm for extended geometries is given in Algorithm 11. The similarities of the geometries are used in the algorithm to define the locations. According to the Algorithm 11, in geometries with even number of actors, two actors are positioned on (x_{sink}, y_{sink}) line with r distance from the sink. If the number of actors is odd, a single actor will be positioned on $(x_{sink}, y_{sink}, z_{sink})$. The rest of the actors are positioned on two planes such as $z_{sink} \pm h$, where h is calculated according to the geometry. On these planes, the actors are distributed with equal angles and two planes are positioned with and angle of $\frac{360}{n-2}$ between them for even number actors and $\frac{360}{n-1}$ between them for odd number of actors.

Kettle [192] showed that the usual molecular orbitals which are used to describe the bonding in the metal cluster may be transformed into the localized two-center and three-center molecular orbitals described by VSEPR theory. When there are more than twelve actors, our system requires multiple sinks to form the actor geometries. Therefore the requirement of our approach is the deployment of more than one sink as the number of actors exceeds twelve.

Algorithm 11 Actor positioning for extended geometries

n: Number of actors
a_i: Actor i
if n is even **then**
  Two actors are positioned on $(x_{sink}, y_{sink}, z_{sink} \pm r)$ line
else
  One actor is positioned on $(x_{sink}, y_{sink}, z_{sink} + r)$
end if
if $n < 11$ **then**
  $\frac{h}{2} = 0.5237$, $a = 1.2156$:
  for $i = 1 \rightarrow 4$ **do**
    $x_{sink} \pm r.a, y_{sink} \pm r.a, z_{sink} - r.\frac{h}{2}$
  end for
  for $i = 5 \rightarrow 6$ **do**
    $x_{sink} \pm r.\frac{a}{\sqrt{2}}, y_{sink}, z_{sink} + r.\frac{h}{2}$
  end for
  For remaining actors: $x_{sink}, y_{sink} \pm r.\frac{a}{\sqrt{2}}, z_{sink} + r.\frac{h}{2}$
else
  $\Phi = 26.565$ and $\Theta = 0°$
  for $i = 1 \rightarrow 5$ **do**
    $x_a = r.cos(\Theta).cos(\Phi)$, $y_a = r.sin(\Theta).cos(\Phi)$, $z_a = r.sin(\Phi)$
    $\Theta = \Theta + 72°$
  end for
  $\Theta = 36°$
  for $i = 6 \rightarrow 10$ **do**
    $x_a = r.cos(\Theta).cos(-\Phi)$, $y_a = r.sin(\Theta).cos(-\Phi)$, $z_a = r.sin(-\Phi)$
    $\Theta = \Theta + 72°$
  end for
end if

7.2.5 Multiple sinks

The scalability of our approach is improved by using multiple sink nodes as another extension of VSEPR theory based method in our system. It has been shown in molecular geometry that the molecules containing multiple central atoms and bonds conform to the general rule of the repulsion among the electron pairs around any central atom. The multiple sink scenario of our approach is modeled as the case with multiple central atoms in the molecular geometries.

Utilization of multiple sinks extends the endurance and scalability of the operation of multiple UAV systems. Since the scenarios with a single sink node use VSEPR theory by forming an analogy to a molecule with a central atom, scenarios with multiple sinks utilize VSEPR theory with an analogy to the connection of multiple molecules. Sinks are larger UAVs with higher payload capacities compared to actors and they are less prone to issues related to weight. While actors are capable of carrying relatively heavier communication hardware as a result of these properties, the lighter payload means the higher altitude and the longer endurance for smaller UAVs [193]. Therefore sinks are used in the aerial network to form the backbone, which is composed of longer communication links. The actors operate with lighter communication hardware by affiliating with a sink and positioning themselves according to VSEPR theory around this sink.

The network of sinks form one of the favored geometries of basic VSEPR theory. For instance, if there are six sinks in an aerial WSAN, they are positioned as the vertices of octahedral without a central node. An example of multiple sink geometries is given in

Fig. 7.11. Four sinks form the tetrahedral geometry with an actor connected to each sink. The sinks are positioned according to the VSEPR theory rules such that each one forming tetrahedral geometries with three actors and a sink.

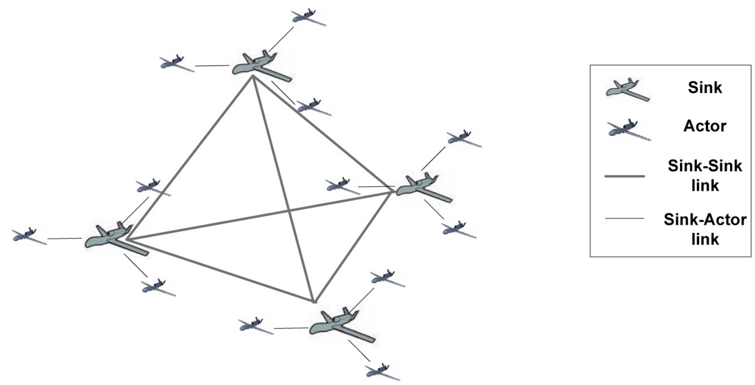


Figure 7.11: An example of multiple sink geometries.

There are three main objectives for excluding the central node in formation of the VSEPR geometries with sinks. First, the communication ranges of sink nodes are larger compared to the actor nodes. Therefore sinks can form a mesh network among themselves, covering an adequately large volume for the mission, without the requirement of a central node with stronger capabilities. Second, introduction of another node type would increase the complexity of the heterogeneous network. Third, the utilization of multiple sinks divides the role of the sink in multiple nodes and prevents the single point of failure. Systems with multiple UAVs operate in highly dynamic environments. The conditions at the beginning of a mission may change during the operation. Therefore the system's ability of adapting to changes in the number of sinks is an important advantage as the number of nodes in the system increases.

The positioning of sinks according to VSEPR theory rules is both challenging and different compared to actor positioning around a central sink. The sinks form a mesh network, which act as the core of the overall UAV system. The defined flight route determines the central point of the geometries and this is shared by all of the sinks. The distances between the sinks change according to the geometries. The edge distances for sink geometries are given in Table 7.1. The transmission range of each sink must be larger than longest edge in the network for a mesh network of sinks.

The sinks form the network by sharing their information with each other. Each of the sinks transmits a network formation packet (NFP) with its ID in the source field and the number of actors connected to it in the payload. The processing of NFP at a sink a is given in Algorithm 12. The sinks record the IDs of the sinks, which they received NFP from, and they calculate the number of the sinks using this information. The sink list is used at a sink for positioning. This list is also saved and updated for future use in case of a change in the sink network such as a dead or an additional sink node. If an NFP is received from a sink, which has a number of actors less than the average, next NFP is loaded with a query for update to this sink. By this way, sinks with less number of actors employ actors from other sinks. Hence, our approach keeps a balanced sink network in terms of the number of actors.

Algorithm 12 Processing of NFP message at sink a

```
n_s: Number of actors for a sink s
S: Sink list kept at the sink a
E: List for unbalanced sinks
Update S
for Each sink i in S do
    Update n_i
    if n_i < ⌊(Σ_{i=1}^{n(S)} n_i) / n(S)⌋ then
        Add i in E
    else if i ∈ E then
        Remove i from E
    end if
end for
if n_a > (Σ_{i=1}^{n_s} n_i) / n_s then
    if E ≠ ∅ then
        for Each sink j ∈ S do
            Send NFP with query for update
        end for
    end if
end if
```

Table 7.1: Distances between edges for sink networks

Geometry	Sink edge distances
Linear	R
Trigonal planar	$R\sqrt{3}$
Tetrahedron	$R\frac{4}{\sqrt{6}}$
Trigonal bipyramid	$R\sqrt{3}$, $R\sqrt{2}$, $2R$
Octahedron	$R\sqrt{2}$, $2R$
Pentagonal dipyramid	$R\sqrt{2}$, $2R$, $R\sqrt{(5-\sqrt{5})/2}$,$R\sqrt{(5+\sqrt{5})/2}$
Square anti-prism	$R\frac{1}{1.645}(2+\frac{1}{\sqrt{2}})$, $R\frac{1}{1.645}(2+\frac{1}{\sqrt{2}})\sqrt{2}$, $R\frac{1}{1.645}\sqrt{1+\sqrt{3}+2\sqrt{2}}$

7.3 Simulation study

7.3.1 Simulation environment

The evaluation of the proposed system is conducted in OPNET modeler [158] by extending the node models created in SOFROP [14]. The transmission radius of a node is taken as 40 meters. There are 30 sensor nodes with IEEE 802.11 MAC layer and random mobility profile.

7.3.2 Simulation results for positioning

7.3.2.1 Coverage

When sensor nodes collect information from the environment, there must be at least one actor in a sensor node's transmission range, which makes the coverage of the network backbone important for the system. The inputs for the volume calculation of actor coverage are the number of spherical coverage volumes, coordinates of the actors, the reception range and expected memory usage by matrix used for modeling spheres. Using these inputs, the union volume of actor coverage is calculated by a numerical approach, which first finds the most distant point in the coordinate system. Then, the real coordinate system is projected to a boolean 3D matrix. The boundary points are found for each sphere and points fitting into the sphere are used to calculate the final volume.

Fig. 7.12 shows coverage for geometries with one sink, two sinks and "3D Deployment" by Lee et al. [194]. Our approach outperforms "3D Deployment" with an average volume difference of 22%. As the number of actors increases up to nine, the coverage of the basic VSEPR theory geometries increase. However it can be observed that the bicapped square antiprism, icosahedron minus one apex and icosahedron are not as effective as the geometries with less actors. Additionally, it is observed that the coverage of 1-sink and 2-sink geometries are similar unless the number of data collectors exceeds seven. Therefore, the number of sinks must be increased to change the geometry of the actors for a more effective coverage when the number of actors exceeds seven.

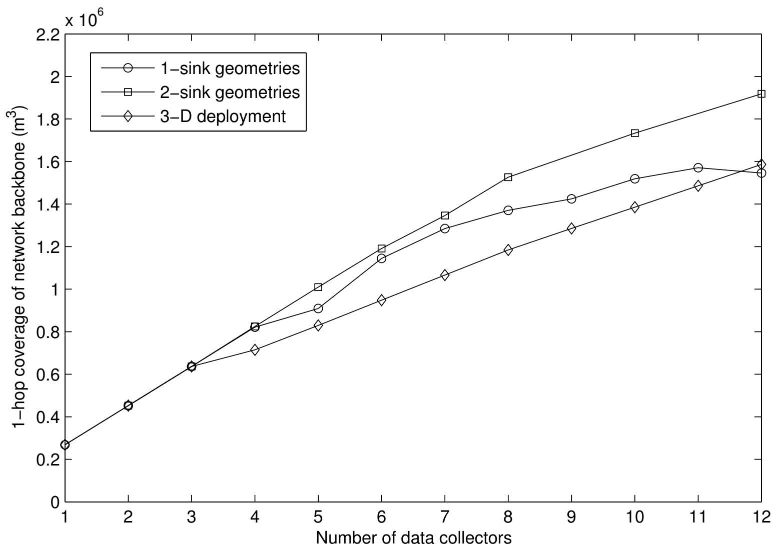


Figure 7.12: 1-hop coverage for different geometries.

In the second set of experiments, the coverage of the proposed VSEPR theory based positioning (VTBP) approach is compared to a partially random positioning (PRP) method. PRP method is designed such that it includes the same number of the sink nodes for each geometry to compare and each actor node is at the same distance to retain the properties of network structure. Fig. 7.13 and Fig. 7.14 show the coverage for a single sink and two sinks geometries of both methods, respectively. The coverage characteristic of our method outperforms PRB in both cases and the performance difference becomes higher as the number of actors increase.

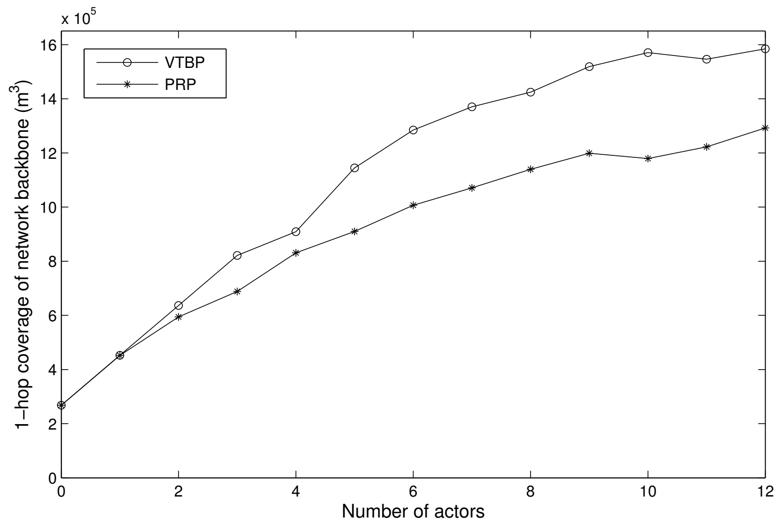


Figure 7.13: 1-hop coverage for our protocol vs. random positioning with 1 sink.

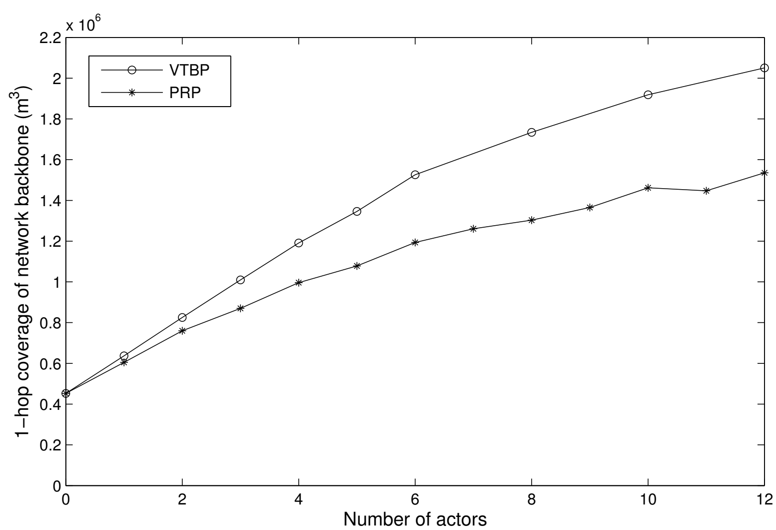


Figure 7.14: 1-hop coverage for our protocol vs. random positioning with 2 sinks.

Fig. 7.15 show the coverage for VSEPR theory based positioning (VTBP) and VSEPR theory based positioning with preferential attachment based actor deployment (VTPA) for increasing number of actors up to eight sinks. In this experiment, the main objective is to see the effects of balanced and preferential attachment based actor deployment in total covered volume for multiple sink geometries. The topologies with the balanced actor deployment have significantly larger coverage as the number of actors is below 50. After the number of actors exceeds 50, the performances of the approaches are very close to each other since the probability of forming different geometries decreases and the topologies become very similar.

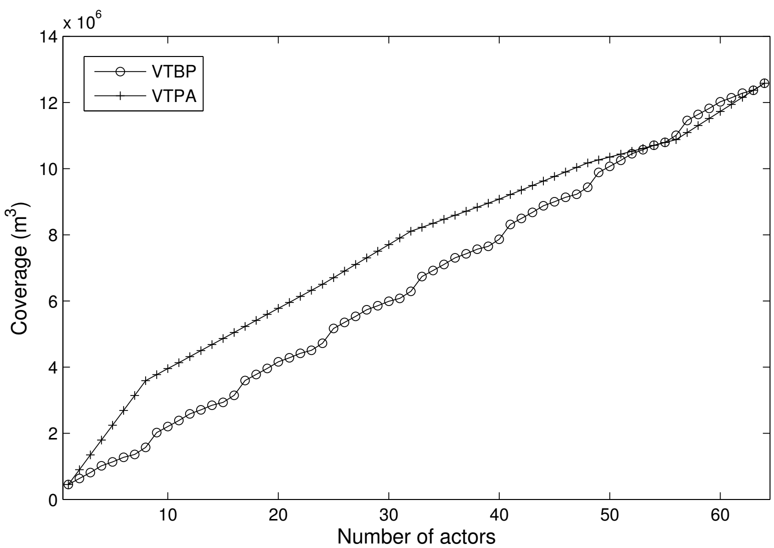


Figure 7.15: 1-hop coverage for VTBP vs. VTPA.

7.3.2.2 Weight

The weight of a sensor UAV decreases by one with each hop it gets further from the actor. The collected information on a sensor UAV can be transmitted to an actor through the path of the sensor UAVs with increasing weight values. Therefore, in contrast to many of the 3D positioning approaches in literature, the coverage of the 3D space is not the only critical criterion to measure the performance of our approach. The sensor measurements can be collected from a large volume of space by utilizing the weight attribute of the sensor nodes. Therefore, we use another metric, average weight value, instead of coverage for the performance assessment of our protocol.

Fig. 7.16 shows the maximum and the minimum weight values averaged over the nodes for all possible geometries. The geometries formed by more actors result in higher average weight values in the network, which means less number of hops for the sensor nodes to transmit the collected information to the actors. The number of unconnected nodes is also decreasing as the geometries become larger. An interesting characteristic of the graph in Fig. 7.16 is the high difference in the average weight between trigonal planar geometry to tetrahedral geometry. Thus, it shows that the geometry gives better performance when more than one plane of actors are used.

The dynamic topology is a fundamental characteristic of our application scenario. Sensor UAVs fly continuously with perturbations in their main flight paths. While the average weight value is critical, the maximum and minimum weight values are also important to assess the suitability of our positioning approach to the mobility of the nodes in our application scenario. The maximum and the minimum hop number of the sensor nodes must not vary among actor areas in a network where the sensor nodes are shared efficiently among actors.

Fig. 7.16 shows that the sensor UAVs are affiliated with the actor UAVs within a smaller range of possible weight values as the number of actor UAVs increase. When the difference between the values of average minimum weight and the average weight values is high, it indicates an ineffective sharing of the nodes as they move in the network. It can be observed that as the geometries evolve, the average minimum weight value increases and the range of weight values that the nodes acquire becomes smaller. Additionally, the performance of the system improves considerably from the trigonal planar geometry to tetrahedral geometry. Therefore the results show that the performance improvement is not only affected by the increase in the number of actors but it also depends on the geometries used.

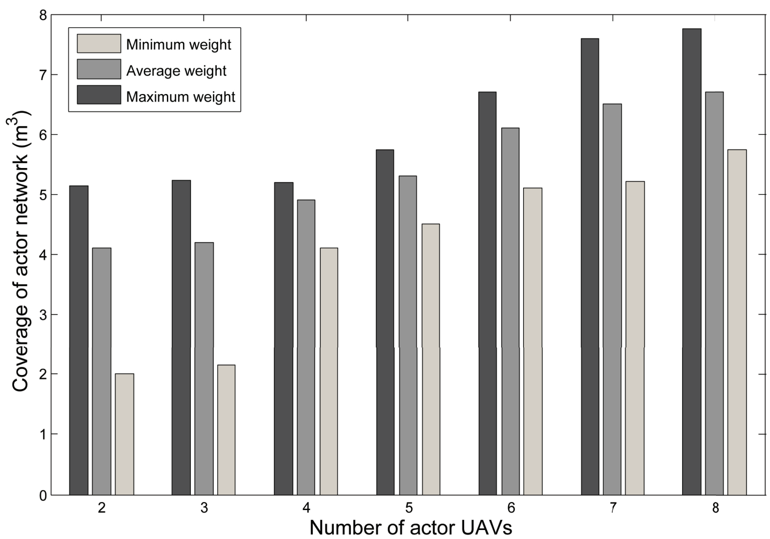


Figure 7.16: Average maximum, minimum and average weight values for all geometries.

7.3.2.3 Cardinality

While using multiple actors, the concurrency becomes essential for an effective utilization of the system. As a result, cardinality is chosen as the metric to evaluate the performance of the system in distributing the actor affiliations. For these scenarios, the sensor nodes move with random mobility in the environment. The average cardinality of the actors are shown in Fig. 7.17 with the range of the collected values. The results show that the average cardinality increases as the number of actors increases. The percentage variation in the cardinalities takes values from 10 to 20% for different geometries. Low fluctuation in the observed values is a result of a balanced sharing of the sensor nodes by the actors in the network.

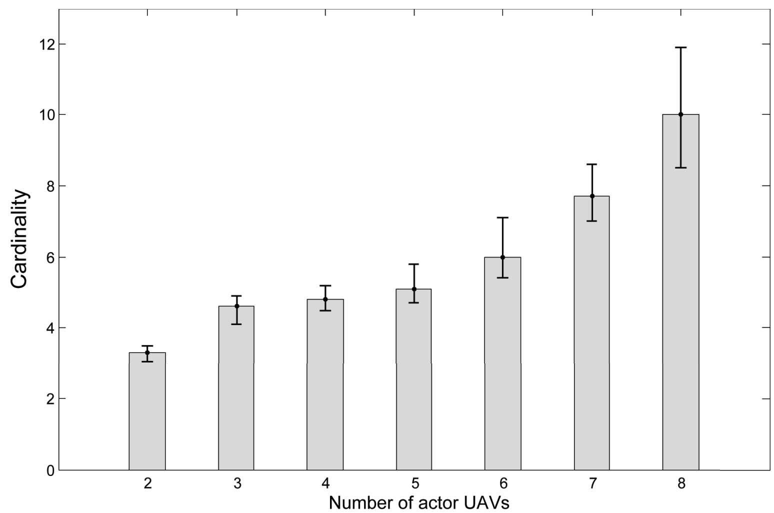


Figure 7.17: Cardinality of actors for different geometries.

7.3.2.4 Betweenness centrality

VSEPR theory is the most successful approach for molecular geometry predictions. Our previous simulations show that our adoption of VSEPR theory results in high performance in coverage. However VSPER theory is not analyzed in terms of the network characteristics of the created geometries. For this analysis, we first use the betweenness centrality, which represents a measure of positional importance. When a node a is in the shortest path between two other nodes, these two nodes depend on the node a for communication. Betweenness centrality values in our application scenario is more important for sinks since all of the actors are the leaves of the network. The betweenness centrality for a sink is the sum of the fraction of all shortest path pairs passing through the sink a, defined as follows:

$$c_B(a) = \sum_{s,t=V} \frac{\sigma(s,t \mid a)}{\sigma(s,t)}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t) paths, and $\sigma(s,t \mid a)$ is the number of those paths passing through a.

We compare the performances for the cases, where the actors are deployed by random deployment, preferential attachment based approach and our balanced approach. Fig. 7.18 shows the average betweenness centrality values of the sinks for geometries with different number of sinks and Fig. 7.19 presents the average deviation of betweenness centrality for the sink nodes. The results given in Fig. 7.18 and Fig. 7.19 show that the preferential attachment based approach has higher values both for average betweenness centrality and the average deviation in betweenness centrality of sink nodes. Fig. 7.19 shows that the average deviation in betweenness centrality decreases for all methods as the number of sinks exceeds three. The value for our approach decreases to one third of its value as the number of sinks increases from three to eight whereas the change in other approaches is about 10% under the same conditions. VSEPR theory based balanced approach provides larger coverage values for all of cases. Therefore, the average deviation in betweenness centrality must be smaller for a better coverage performance in our approach.

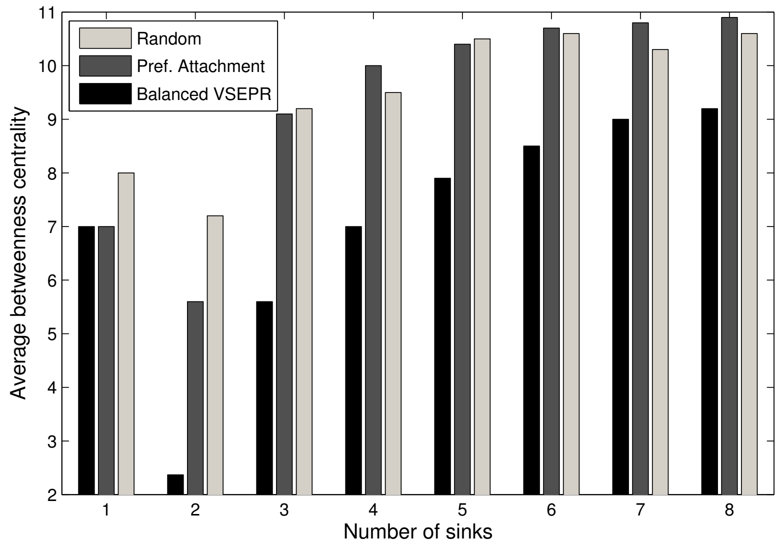


Figure 7.18: Average betweenness centrality of sink nodes.

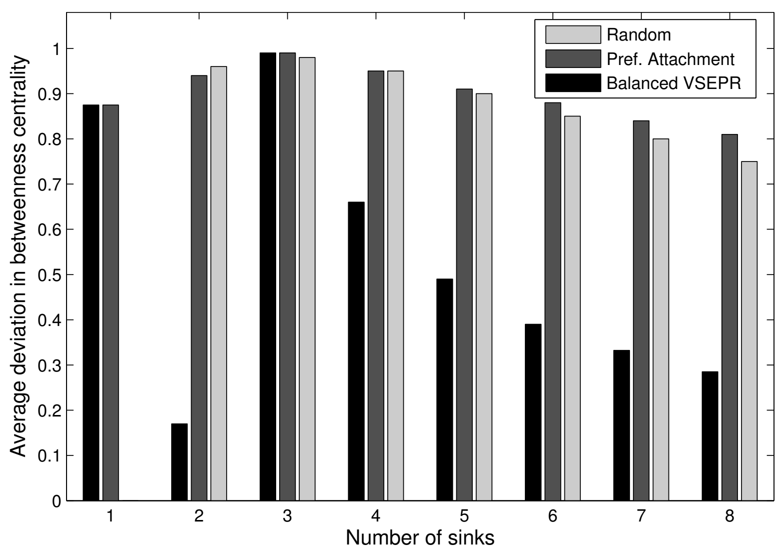


Figure 7.19: Average deviation in betweenness centrality of sink nodes.

7.3.2.5 Clustering coefficient

Another metric we use to analyze the network characteristics of our approach is the clustering coefficient (CC). We compare the CC values of the sinks for the geometries formed by the deployment of actors based on random deployment, preferential attachment based approach and our balanced approach. The network graph formed by the VSEPR topologies are unweighted. Thus, the CC of a node u is the fraction of possible triangles through that node, which is defined as follows:

$$c_u = \frac{2T(u)}{deg(u)(deg(u) - 1)}$$

where $T(u)$ is the number of triangles through node u and $deg(u)$ is the degree of u.

Fig. 7.20 shows the average sink CC for different number of sinks. The sink CC increases as the number of sinks in the geometry is increased. The preferential attachment based approach has the highest and the VSPER-based balanced approach has the lowest sink CC values for all of the geometries whereas the values for random positioning are in between the other two approaches.

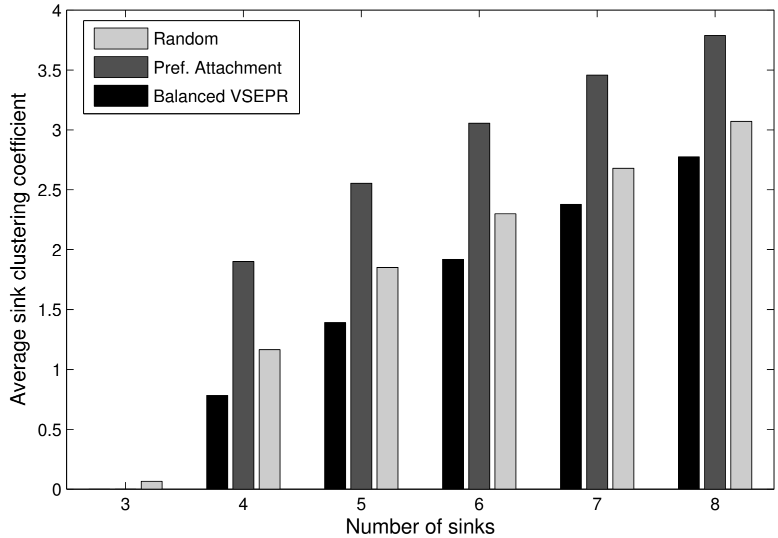
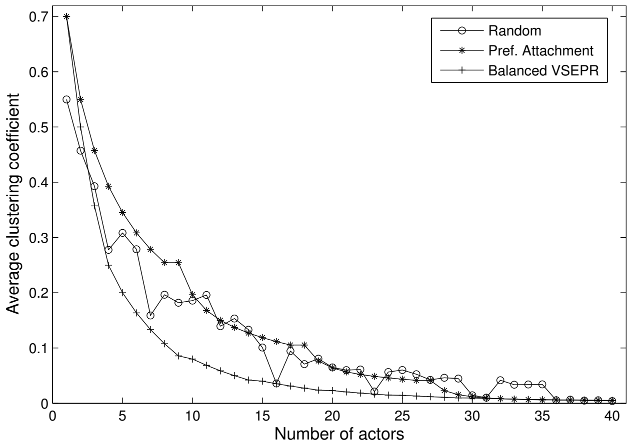
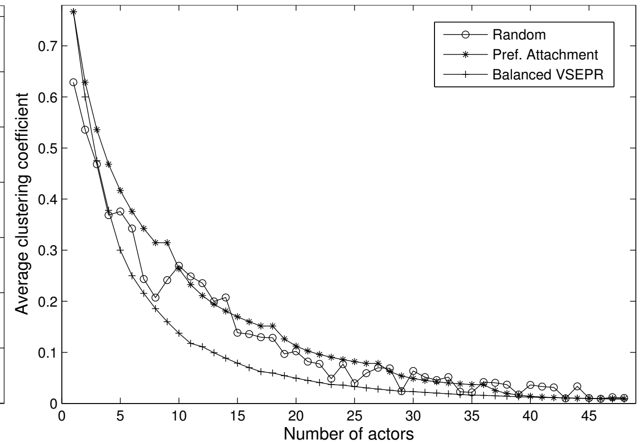


Figure 7.20: Average sink clustering coefficient for different number of sinks.

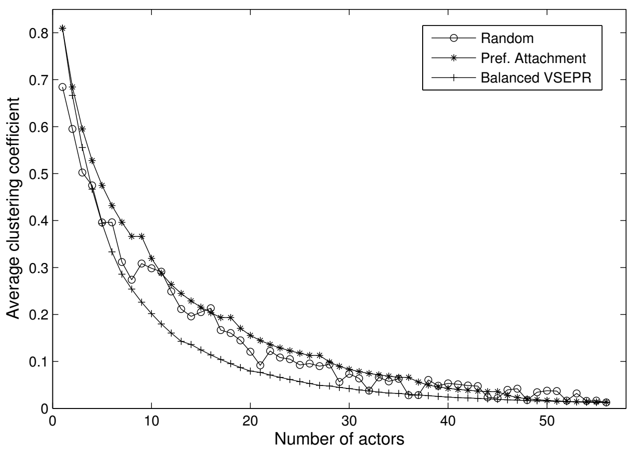
Fig. 7.21 shows the average CC values of actors for different number of sinks. For all of the cases, balanced VSEPR-based approach has smaller average CC compared to preferential attachment based approach. Random positioning method has values in between the other two approaches most of the time.



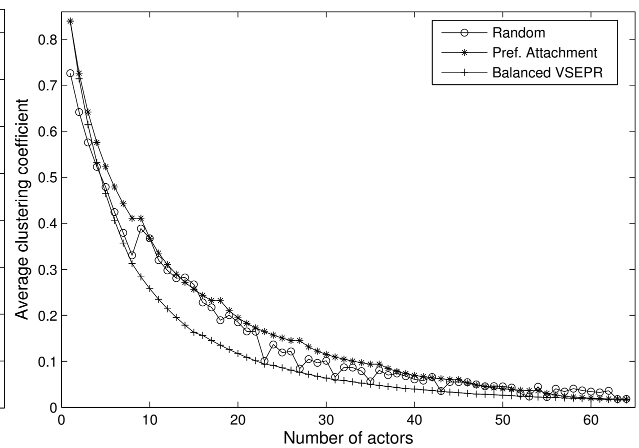
(a) Topologies with five sinks.



(b) Topologies with six sinks.



(c) Topologies with seven sinks.



(d) Topologies with eight sinks.

Figure 7.21: CC values of actors for different number of sinks.

The results given in Fig. 7.20 and Fig. 7.21 show that the preferential attachment based approach has higher CC values both for actors and the sinks. However VSEPR theory based balanced approach has a larger coverage for all of different cases. Therefore, the results indicate that the coverage of UAV network is inversely proportional to the clustering in our approach.

7.3.3 Simulation results for positioning with hybrid antenna

The simulation study is conducted in the ns-2 simulator. The O-BESPAR antenna model, VSEPR theory flight structures, the communication and flight control protocols are implemented and tested. The performance of Packet Reception Ratio (PRR), actors' reorganization delay, and RSSI are evaluated to show the efficiency of the algorithm. The relationship between RSSI and coverage by two antenna models is also discussed.

There are 30 UAVs in the simulation, two to eight of which are actors and one of which is the sink. Zigbee [195] which has been integrated in many off-the-shelf sensors on UAV is implemented for MAC and physical layer communications. The target area size is $1000m \times 1000m$. According to the specs of UAVs [196], the simulation time is set to 15 minutes and the flying speed of actor and sink is 1 m/s. The sink and actors fly with a predefined plan while maintaining the geometry. We assume the sensor UAVs fly at altitudes different from the actors and the sink. Their flying speed and movements are random and their communications cause interference to the UAVs in geometrical flight. Five beacons per second are sent out by the actor to search for the sink. Each UAV includes a queue of 50 packets.

The omni-directional and O-BESPAR models work in 2.4GHz frequency band. The sink and actor UAVs have the same antenna structures. The transmission radius of omnidirectional antenna on each UAV is 10 meters. According to the transmission range, relationship between omni and directional modules [189], the transmission distance of the beamforming is 28 meters.

For bi-directional beam sweeping, the beamwidth is 60°. Therefore the angle of each sweeping is 60° and the beam hops from 0 to 180°.

7.3.3.1 Packet transmission experiment

The PRR of two antennas are evaluated for all of the VSEPR geometries. Fig. 7.22 and 7.23 show the PRR performance of the omni and O-BESPAR antennas in different geometries when the number of actors increases from two to eight. When the number of actors are lower than six, the PRR of omni antenna is between 50% and 60%. The PRR of O-BESPAR antenna varied between 97% and 99%. In pentagonal bipyramid and square antiprismatic geometries, the PRR of actor UAVs drops between 40% and 50% by using omni antenna. Meanwhile, O-BESPAR antenna guarantees the PRR of AWSANs higher than 95%. The packet loss of omnidirectional antenna increases as more actors transmit packets to the sink. There are two fundamental reasons for packet loss. Due to the interference from other non-actor UAVs in flight, omni antenna has much more transmission collisions than O-BESPAR antenna. In addition, the actor has varied link quality values as the RSSI is affected by antenna orientation. The PRR of actor 1 drops in square antiprismatic geometry because of the poor RSSI at that orientation. For O-BESPAR antenna, the increase of number of actors makes directional module of the sink busy. The data packet is lost if the buffer is full or the timestamp is expired.

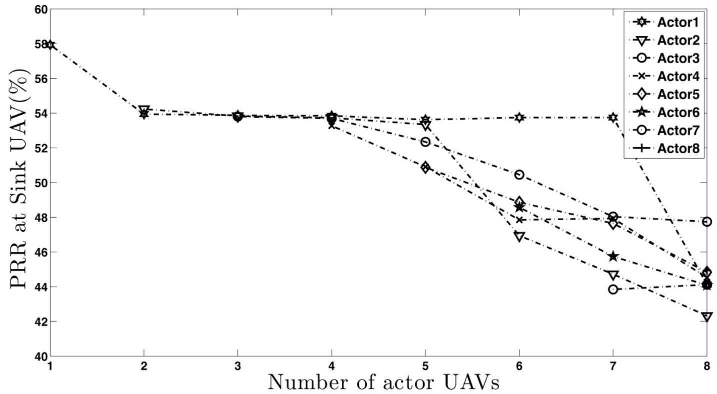


Figure 7.22: PRR of Omni-directional antenna model in different geometries.

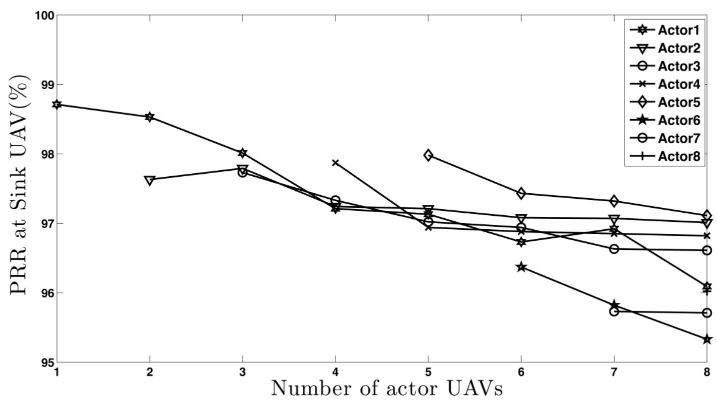


Figure 7.23: PRR of O-BESPAR antenna model in different geometries.

7.3.3.2 Reorganization experiment

The actor-sink link is prone to failure due to flying path dynamics and interference. Then the sink repositions actors and updates the geometry for a better coverage. This process is called reorganization. In this experiment, UAV actor 1 leaves the network during the flight. Then, sink uses Algorithm 10 and changes the flight geometry. The positions of the actors must be updated with the PUM message for transitioning to the new geometry. The time delay is defined as the duration between the actor 1 leaving and all actors receiving PUM message.

Fig. 7.24 presents the reorganization time delay of omni and O-BESPAR antennas in different geometries.

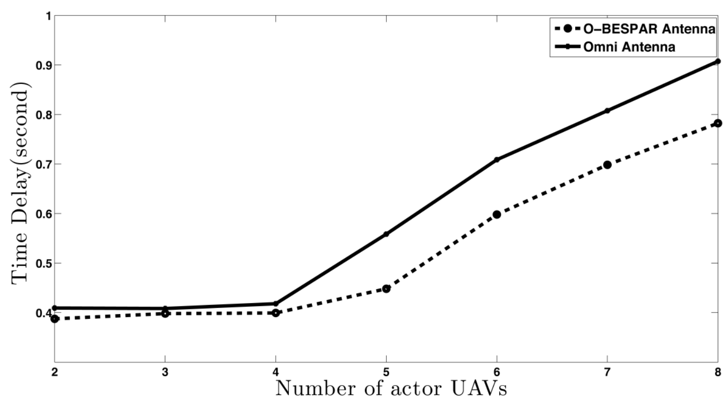


Figure 7.24: Reorganization time delay of antenna models

As Fig. 7.24 shows, larger number of actors results in longer reorganization delay. Generally, the time delay of both antenna models is similar since O-BESPAR also uses omni antenna module to search neighboring UAVs. However, there is a time gap between two models. For omni antenna, as a result of transmission of both control and data packets, either actor or sink does not exchange beacons until the end of data transmission. It causes a long reception delay of control packets. For O-BESPAR antenna, only the beacon packets are sent through omni module, which minimizes the time delay of actor relocation.

7.3.3.3 Antenna orientation and coverage

As discussed in the previous section, the orientation of omni antenna module causes poor RSSI which decreases PRR at some positions. The loss of beacon messages increases reorganization time delay of UAVs. Therefore, those UAVs with poor RSSI need to fly closer to the sink in order to guarantee the RSSI. However, as distance between the actor and the sink becomes shorter, network coverage becomes smaller. As a result, there is a tradeoff between coverage and RSSI of UAVs.

Fig. 7.25 shows the RSSI of each actor UAV in the geometry flight.

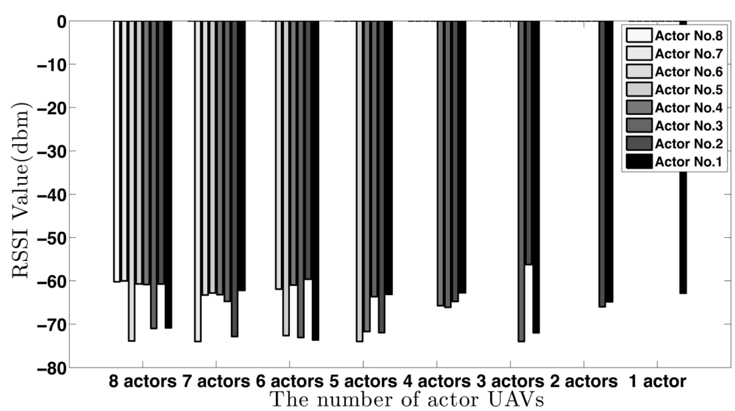


Figure 7.25: RSSI values of the omni antenna in different geometry

Most of actors maintain the RSSI in each geometry, however, some actors have higher beacon packet loss due to poor RSSI of the orientation. Based on the RSSI values in different geometries, the actor whose RSSI is smaller than the original flies closer to the sink to achieve the same RSSI.

When the actors fly closer to the sink, the coverage of network is changed. The coverage is measured by the volume covered by the network geometry in the air. Fig. 7.26 presents the original and updated network coverage values. Total coverage of the network is critical since when it increases, the data sensing range of the actors also increases. The network has to sacrifice at most 13% of coverage to fulfill the RSSI requirement of omni antenna. In particular, this tradeoff between RSSI and network coverage can be used as an indication of the network design, such as the requirements of delay tolerant UAV networks.

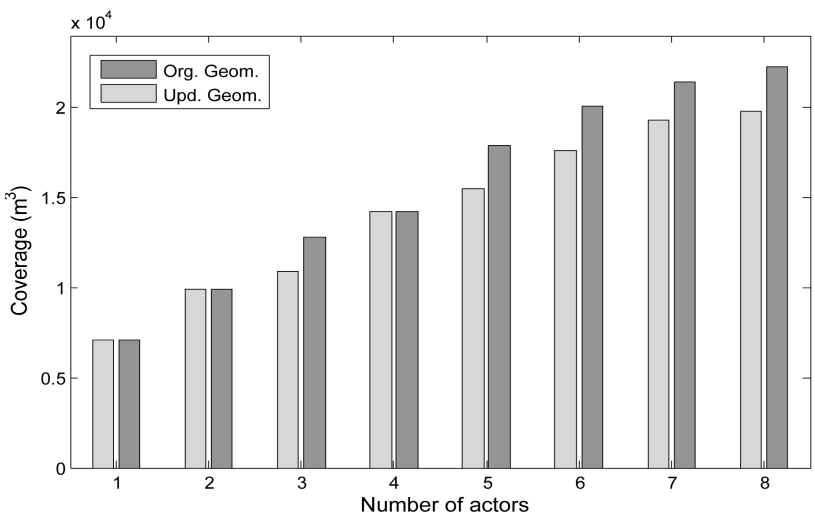


Figure 7.26: The UAV coverage of each geometry flight

CHAPTER 8
CONCLUSIONS AND FUTURE WORK

The work described in this dissertation presents our contributions to the fields of routing, localization and positioning in WSANs, focusing on particular application scenarios. The proposed approaches are evaluated in the OPNET modeler by designing wireless sensor and actor node models.

We first introduced LRP-QS, a routing protocol providing service differentiation in stationary WSANs with a low computational complexity at sensor nodes. In LRP-QS, the network is organized so that each actor forms an actor area composed of multiple sensor nodes. Sensor nodes collect information from the environment, the type and importance of which depend on the interests distributed by the sink. The interests have initial weight values, which is dynamically updated according to the changes in the observed values of the events of interest as they occur. Data packets carry rate values on their path to the actors and they are dropped probabilistically at sensor nodes. The results of the simulation study verify the effectiveness of the proposed scheme compared to QBRP in terms of packet loss, memory consumption and control overhead. We also used a nonsensitive ranking method for dynamic weight assignment to the interests and modified the method according to our requirements. The weights of the interests in LRP-QS are altered according to the changes in the observed values of the events. The simulations show that LRP-QS can dynamically prioritize the interests with a relatively low number of updates in the network.

As a future extension of LRP-QS, the effects of the ranking parameters on the routing protocol can be analyzed in more detail to tailor the algorithm for other possible scenarios. The trade-off between the range for producing an update and the number of updates for interests depends on the network type and the requirements. Formulization of the relation of these two parameters is another possible extension.

We integrated the routing strategy of LRP-QS with a hierarchical network organization method and proposed SOFROP for routing in WSANs with mobile sensor nodes. SOFROP is designed for Amazon scenario, which has peculiarities such as the actor nodes remain static but irregularly deployed while the sensor nodes are moving in an unpredictable pattern. SOFROP network organization builds a structured network topology that permanently adapts according to the river dynamics. Simulation results verify the effectiveness of the proposed scheme in keeping the service differentiation behavior of LRP-QS in the unprecedented conditions of the chosen scenario.

Additional application scenario specific characteristics, such as the direction of the river, can be incorporated into the network organization algorithm as a future direction. Another possible extension of the same idea would be increasing the energy-efficiency of actors by sleep and active cycles, depending on the location of majority of sensor nodes in the river. Although the initial assumption is actors are more powerful, they can be also on batteries, which would require efficient energy utilization.

For localization of events in Amazon river scenario, we adapted the network organization in SOFROP for affiliation of sensor nodes with multiple actors. The multi hop localization protocol aim to improve the on-site monitoring with a scalable approach. The particular scenario has its own challenges, which require high adaptability to failures and high mobility of sensor nodes. The proposed localization algorithm overcomes these challenges by a locality preserving approach complemented with an idea that benefits from the motion pattern of the sensors. The algorithm aims to retrieve location information at the actors rather than the sensor nodes and it adopts 1-hop localization approach in order to address the limited lifetime of the WSAN. The selection of a realistic mobility model is critical for performance evaluation of a localization algorithm. Therefore, a subsurface current mobility model is adopted and tailored according to the requirements of the scenario. Through extensive simulations, we have shown that the localization estimation can be realized using local multi-hop information. In overall, as the multi-hop chains are allowed to become longer, more positions can be estimated with the cost of lower accuracy. The selection of the maximum hop number is therefore an issue depending on the requirements of network.

As future work, our localization approach can be integrated with an actor positioning strategy since the actor positions and the communication in the network among actors affect the performance of localization. Additionally, the routing and energy consumption can be improved by using the localization information for data aggregation and dissemination. The accuracy of the proposed algorithm can be further improved with RSS or other measurement techniques at the expense of increased energy consumption.

After the Amazon River scenario, we explored an animal monitoring application and introduced new network formation, mobility and role determination algorithms to provide a complete primate group model. These methods are based on Lévy walk, center of mass and preferential attachment concepts. The preferential attachment model is extended for the formation and mobility models according to the characteristics of the animal groups. These models are also extendible with properties specific to observed animal species. The social monitoring algorithm for deciding on the roles of primates uses the observed spatial-temporal interaction patterns. In the application scenario, each animal is equipped with a sensor node for the monitoring of the social system. In the case of gorillas, it is shown how the data about the social structure in general aids in the design of an efficient protocol for capturing the social network of a group. Simulation results show that the outputs of preferential attachment based node formation and mobility models match with the characteristics of the animal groups under consideration. The approach in primate role determination is also adapted for a similar problem in networks of human beings. An interaction evaluation function, a ranking and a grouping method are designed to automatically generate the social network of an individual by analyzing and assessing smartphone usage and interaction data.

Natural extensions of our primate monitoring system include application and modification of the proposed algorithms to different animal groups. As another future direction, role determination algorithm can be extended for human social networks with hierarchical structures such as an office environment. The future direction for social network generation approach would be the determination of specific interaction weight values for specific social groups.

A possible extension to the proposed protocols would be integrating them with an actor positioning approach. Since positioning has been frequently studied in 2D, we explored a relatively unexplored topic and proposed APAWSAN, an actor positioning method for aerial WSANs. APAWSAN is designed to improve the on-site monitoring of the plume in a volcanic eruption. In this scenario, the VSEPR theory is utilized to position one central and multiple peripheral actors. The basic rules of VSEPR theory are extended to overcome the limitation on the number of actors and only local communication is used for actor positioning. The dynamic positioning requirement, defective signal strength of omnidirectional antenna and unreliability of links pose challenges for AWSANs. Our algorithms take advantage of the hybrid antenna model to improve the efficiency and availability. We present extensive simulations with 3D radio characterization to demonstrate the improvement of PRR and network reorganization delay. The variance of network coverage caused by omnidirectional antenna orientation is also discussed. The experiments show that the system provides better coverage than a partially random positioning while keeping 1-hop connectivity between each actor and the sink.

As future work, O-BESPAR antenna module can be built to test and observe the performance of our approach in a real 3D testbed. Another possible future work would be the extension of the application of molecular geometry to the actor positioning. Molecular structure formations can be analyzed for large clusters of atoms to improve the scalability of APAWSAN.

LIST OF REFERENCES

[1] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design.* Academic Press, January 2008.

[2] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, pp. 351–367, October 2004.

[3] M. C. Vuran, D. Pompili, and T. Melodia, *Future Trends in Wireless Sensor Networks.* John Wiley and Sons, Inc., 2008.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292–2330, August 2008.

[5] D. Peleg, *Distributed computing: A locality-sensitive approach.* Society for Industrial and Applied Mathematics, 2000.

[6] F. Xia, "QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks," *Sensors*, vol. 8, pp. 1099–1110, June 2008.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 2–16, Feb. 2003.

[8] N. Patwari, J. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," in *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, 2005.

[9] F. Regan, A. Lawlor, B. Flynn, J. Torres, R. Martinez-Catala, C. O'Mathuna, and J. Wallace, "A demonstration of wireless sensing for long term monitoring of water quality," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pp. 819–825, December 2009.

[10] B. Turgut and R. P. Martin, "Using a-priori information to improve the accuracy of indoor dynamic localization," in *Proceedings of the ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 394–404, October 2009.

[11] M. L. Morrison, *Wildlife restoration: techniques for habitat analysis and animal monitoring.* Island Press, 2002.

[12] M. İ. Akbaş and D. Turgut, "Lightweight Routing with QoS Support in Wireless Sensor and Actor Networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, IEEE, 2010.

[13] M. İ. Akbaş and D. Turgut, "Lightweight Routing with Dynamic Interests in Wireless Sensor and Actor Networks," *Elsevier Ad Hoc Networks*, vol. 11, pp. 2313–2328, November 2013.

[14] M. İ. Akbaş, M. R. Brust and D. Turgut, "SOFROP: Self-organizing and fair routing protocol for wireless networks with mobile sensors and stationary actors," *Elsevier Journal of Computer Communications*, vol. 34, pp. 2135–2146, 2011.

[15] M. İ. Akbaş, M. R. Brust and D. Turgut, "SOFROP: Self-Organizing and Fair Routing Protocol for Wireless Networks with Mobile Sensors and Stationary Actors," in *Proceedings of the IEEE Local Computer Networks (LCN)*, pp. 456–463, October 2010.

[16] M. İ. Akbaş, M. R. Brust and D. Turgut, "Local positioning for environmental monitoring in wireless sensor and actor networks," in *Proceedings of the IEEE Local Computer Networks (LCN)*, pp. 806–813, IEEE, 2010.

[17] M. R. Brust, M. İ. Akbaş, and D. Turgut, "Multi-hop Localization System for Environmental Monitoring in Wireless Sensor and Actor Networks," *Wiley Journal of Concurrency and Computation: Practice and Experience*, vol. 25, pp. 701–717, April 2013.

[18] M. İ. Akbaş, M. Erol-Kantarci and D. Turgut, "Localization for Wireless Sensor and Actor Networks with Meandering Mobility," *Under review in Transactions on Computers*, 2013.

[19] M. İ. Akbaş, M. R. Brust, C. H. C. Ribeiro and D. Turgut, "Deployment and Mobility for Animal Social Life Monitoring Based on Preferential Attachment," in *Proceedings of the IEEE Local Computer Networks (LCN)*, pp. 488–495, IEEE, 2011.

[20] M. İ. Akbaş, M. R. Brust, C. H. C. Riberio and D. Turgut, "fAPEbook - Animal Social life Monitoring with Wireless Sensor and Actor Networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–5, December 2011.

[21] M. İ. Akbaş, M. R. Brust, D. Turgut and C. H. C. Ribeiro, "Modeling Primate Social Life Monitoring with Wireless Sensor and Actor Networks," *Under review in Elsevier Computer Networks*, 2013.

[22] M. İ. Akbaş, R. N. Avula, M. Bassiouni and D. Turgut, "Social Network Generation and Friend Ranking Based on Mobile Phone Data," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 37–41, June 2013.

[23] M. İ. Akbaş and D. Turgut, "APAWSAN: Actor positioning for aerial wireless sensor and actor networks," in *Proceedings of the IEEE Local Computer Networks (LCN)*, pp. 567–574, October 2011.

[24] M. İ. Akbaş, G. Solmaz and D. Turgut, "Actor Positioning Based on Molecular Geometry in Aerial Sensor Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 508–512, June 2012.

[25] M. İ. Akbaş, G. Solmaz and D. Turgut, "Molecular Geometry Inspired Positioning for Aerial Networks," *Under review in the Transactions on Vehicular Technology*, 2013.

[26] K. Li, M. İ. Akbaş, D. Turgut, S. S. Kanhere and S. Jha, "Reliable Positioning with Hybrid Antenna Model for Aerial Wireless Sensor and Actor Networks," in *Under submission in the IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014.

[27] A. Boukerche, R. B. Araujo, and L. Villas, "A novel QoS based routing protocol for wireless actor and sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom)*, pp. 4931–4935, November 2007.

[28] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, pp. 2826–2841, October 2007.

[29] W. Heinzelman and A. C. snd H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.

[30] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1948–1953, March 2003.

[31] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: Two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, pp. 161–175, January 2005.

[32] M. Chatterjee, S. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, pp. 193–204, April 2002.

[33] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, pp. 955–972, July 2009.

[34] K. Robinson, D. Turgut, and M. Chatterjee, "Entropy based Clustering in Mobile Ad Hoc Networks," in *Proceedings of IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–5, April 2006.

[35] D. Turgut, K. Robinson, and M. Chatterjee, "Entropy based Clustering in Mobile Ad Hoc Networks," *Journal of Ubiquitous Computing and Intelligence (JUCI)*, vol. 1, pp. 101–109, April 2007.

[36] N. Aydin, F. Nait-Abdesselam, V. Pryyma, and D. Turgut, "Overlapping clusters algorithm in ad hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom)*, pp. 1–5, December 2010.

[37] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks," in *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA)*, August 2004.

[38] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "Extending Network Life by Using Mobile Actors in Cluster-based Wireless Sensor and Actor Networks," in *International Federation for Information Processing (IFIP), Wireless Sensor and Actor Networks II*, vol. 264, pp. 74–84, May 2008.

[39] L. Zhang, Z. Hu, Y. Li, and X. Tang, "Grouping-Based Clustering Routing Protocol in Wireless Sensor Networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, pp. 2452–2455, September 2007.

[40] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 258–271, July 2004.

[41] J. Lin and M. Liao, "A clustering patch hierarchical routing protocol for wireless sensor networks," in *International Conference on Computer Science and Education (ICCSE)*, pp. 941–948, August 2010.

[42] A. Boukerche, A. Martirosyan, and R. Pazzi, "An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks," *Mobile Networks and Applications*, vol. 13, pp. 614–626, December 2008.

[43] S. Basagni and E. Jonsson, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proceedings of International Vehicular Technology Conference (VTC)*, pp. 889–893, September 1999.

[44] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in Ad Hoc and Hybrid Networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1079–1088, November 2002.

[45] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P. V. Mieghem, S. Avallone, G. Ventre, P. Aranda-Gutirrez, M. Hollick, R. Steinmetz, L. Iannone, and K. Salamatian, "Research challenges in QoS routing," *Computer Communications*, vol. 29, pp. 563 – 581, March 2006.

[46] M. A. El-Gendy, A. Bose, and K. G. Shin, "Evolution of the Internet QoS and support for soft real-time applications," in *Proceedings of the IEEE*, vol. 91, pp. 1086–1104, July 2003.

[47] Y. Li, C. S. Chen, Y.-Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: A survey," in *International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, pp. 159–164, November 2007.

[48] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 201–205, September 2004.

[49] E. Cañete, J. Chen, M. Díaz, L. Llopis and B. Rubio, "A service-oriented approach to facilitate WSAN application development," *Ad Hoc Networks*, vol. 9, no. 3, pp. 430–452, 2011.

[50] Y. Tao, Y. Zhang, and Y. Ji, "Flow-balanced routing for multi-hop clustered wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 541–554, 2013.

[51] T. Melodia, D. Pompili, V. Gungor, and I. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 1116–1129, October 2007.

[52] G. A. Shah, M. Bozyigit, O. B. Akan, and B. Baykal, "Real-time coordination and routing in wireless sensor and actor networks," in *Proceedings of Next Generation Teletraffic and Wired/Wireless Advanced International Conference (NEW2AN)*, May 2006.

[53] E. Ngai, Y. Zhou, M. R. Lyu, and J. Liu, "A delay-aware reliable event reporting framework for wireless sensor-actuator networks," *Ad Hoc Networks*, vol. 8, pp. 694–707, September 2010.

[54] A. Sama and K. Akkaya, "Real-time routing for mobile sensor/actor networks," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pp. 821–828, October 2008.

[55] K. L. Hung, B. Bensaou, J. Zhu, and F. Nait-Abdesselam, "Energy-Aware Fair Routing in Wireless Sensor Networks with Maximum Data Collection," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 8, pp. 3438–3443, December 2006.

[56] K. Morita, K. Ozaki, N. Hayashibara, T. Enokido, and M. Takizawa, "Evaluation of Reliable Data Transmission Protocol in Wireless Sensor-Actuator Network," in *Proceedings of Advanced Information Networking and Applications Workshops (AINAW)*, vol. 2, pp. 713–718, May 2007.

[57] V. Paruchuri, A. Durresi, and L. Barolli, "Energy aware routing protocol for heterogeneous wireless sensor networks," in *Proceedings of the Database and Expert Systems Applications*, pp. 133–137, August 2005.

[58] D. Zhang, G. Li, and K. Zheng, "An energy-balanced routing method based on forward-aware factor for Wireless Sensor Network," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2013.

[59] T. Houngbadji and S. Pierre, "QoSNET: An integrated QoS network for routing protocols in large scale wireless sensor networks," *Computer Communications*, vol. 33, pp. 1334–1342, July 2010.

[60] B. C. Villaverde, S. Rea, and D. Pesch, "InRout - A QoS aware route selection algorithm for industrial wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 458–478, 2012.

[61] M. Baunach, "Collaborative memory management for reactive sensor/actor systems," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pp. 953–960, October 2010.

[62] O. Banimelhem and S. Khasawneh, "GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1346–1361, 2012.

[63] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 849–857, August 2010.

[64] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance," *Information Fusion*, no. 0, pp. –, 2013.

[65] W. N. Colley, "Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained." http://www.colleyrankings.com/matrate.pdf, September 2003.

[66] K. Massey, "Statistical Models Applied to the Rating of Sports Teams, Master's thesis, Bluefield College, Bluefield," 1997.

[67] L. Ingram, "Ranking NCAA sports teams with linear algebra, master's thesis, College of Charleston," 2007.

[68] A. Y. Govan, A. N. Langville, and C. D. Meyer, "Offense-defense approach to ranking team sports," *J. Quant. Anal. Sports*, vol. 5, pp. 1–17, 2009.

[69] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings, "Sensitivity and Stability of Ranking Vectors," *SIAM Journal on Scientific Computing*, vol. 33, pp. 1077–1102, May 2011.

[70] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[71] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 201–205, September 2004.

[72] B. D. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, "Graphical properties of easily localizable sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 177–191, 2009.

[73] J. Jeong, S. Guo, T. He, and D.-C. Du, "Autonomous Passive Localization Algorithm for Road Sensor Networks," *IEEE Transactions on Computers*, vol. 60, no. 11, pp. 1622–1637, 2011.

[74] H. Shi, X. Li, Y. Shang, and D. Ma, "Error analysis of quantised RSSI based sensor network localization," *International Journal of Wireless and Mobile Computing*, vol. 4, no. 1, pp. 31–40, 2010.

[75] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using Cell-ID sequence matching," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 293–306, June 2011.

[76] D. Niculescu and B. Nath, "DV Based Positioning in Ad hoc Networks," *Journal of Telecommunication Systems*, vol. 22, no. 8, pp. 267–280, 2003.

[77] Y. Wang, S. Lederer, and J. Gao, "Connectivity-Based Sensor Network Localization with Incremental Delaunay Refinement Method," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2401–2409, April 2009.

[78] B. Turgut and R. P. Martin, "Restarting particle filters: an approach to improve the performance of dynamic indoor localization," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, November-December 2009.

[79] B. Turgut and R. P. Martin, "A multi-hypothesis particle filter for indoor dynamic localization," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pp. 742–749, October 2009.

[80] B.-H. Liu, M.-L. Chen, and M.-J. Tsai, "Message-Efficient Location Prediction for Mobile Objects in Wireless Sensor Networks Using a Maximum Likelihood Technique," *IEEE Transactions on Computers*, vol. 60, no. 6, pp. 865–878, 2011.

[81] H. A. B. F. Oliveira, A. Boukerche, E. F. Nakamura, and A. A. F. Loureiro, "An Efficient Directed Localization Recursion Protocol for Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 58, pp. 677–691, May 2009.

[82] H. Xu, W. Liu, and B. Wang, “Subarea Localization Performance of The Divide-and-Cover Node Deployment in A Long Bounded Belt Scenario,” *IEEE Transactions on Computers*, vol. PrePrints, no. 99, 2013.

[83] J. Maneesilp, C. Wang, H. Wu, and N.-F. Tzeng, “RFID Support for Accurate 3D Localization,” *IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1447–1459, 2013.

[84] J. Chen, C. Wang, Y. Sun, and X. S. Shen, “Semi-supervised Laplacian regularized least squares algorithm for localization in wireless sensor networks,” *Computer Networks*, vol. 55, no. 10, pp. 2481–2491, 2011.

[85] B. Krishnamachari and S. Iyengar, “Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[86] I. Getting, “The global positioning system,” *IEEE Spectrum*, vol. 30, pp. 36–47, December 1993.

[87] L. Yang and K. Ho, “An approximately efficient TDOA localization algorithm in closed-form for locating multiple disjoint sources with erroneous sensor positions,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4598–4615, 2009.

[88] T. Eren, “Cooperative localization in wireless ad hoc and sensor networks using hybrid distance and bearing (angle of arrival) measurements,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–18, 2011.

[89] B. Turgut and P. Martin, “Localization for indoor wireless networks using minimum intersection areas of iso-RSS lines,” in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pp. 962–972, October 2007.

[90] H. Ahn, Y.-H. Lee, H.-J. Cho, S.-B. Rhee, and J.-H. Lee, “A RSSI-Based Approach for Localization of Wireless Sensor Network in Indoor,” in *Proceedings of the International Conference on IT Convergence and Security (ICITCS)*, pp. 123–127, December 2012.

[91] S. Rallapalli, L. Qiu, Y. Zhang, and Y.-C. Chen, “Exploiting temporal stability and low-rank structure for localization in mobile networks,” in *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, pp. 161–172, September 2010.

[92] H. Wang, S. Sen, A. Elgohary, M. Elgohary, M. Youssef, and R. R. Choudhury, “No need to war-drive: unsupervised indoor localization,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 197–210, June 2012.

[93] M. Michaelides and C. Panayiotou, “SNAP: Fault Tolerant Event Location Estimation in Sensor Networks Using Binary Data,” *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1185–1197, 2009.

[94] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc wireless sensor networks," in *Proceedings of Annual International Conference on Mobile Computing and Networking*, pp. 166–179, July 2001.

[95] C. Savarese, J. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proceedings of the Annual USENIX Technical Conference*, pp. 317–327, June 2002.

[96] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," in *Proceedings of the International Workshop on Information Processing in Sensor Networks*, pp. 333–348, 2003.

[97] J. Fang, M. Cao, A. S. Morse, and B. D. O. Anderson, "Sequential Localization of Sensor Networks," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 321–350, 2009.

[98] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "A novel location-free greedy forward algorithm for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 2096–2101, May 2008.

[99] A. Caruso, F. Paparella, L. Vieira, M. Erol, and M. Gerla, "The meandering current mobility model and its impact on underwater mobile sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 13–19, April 2008.

[100] M. Erol, L. Vieira, A. Caruso, F. Paparella, M. Gerla, and S. Oktug, "Multi Stage Underwater Sensor Localization Using Mobile Beacons," in *Proceedings of the Workshop on Under Water Sensors and Systems (in conjunction with SENSORCOMM)*, pp. 25–31, August 2008.

[101] J. K. Hart and K. Martinez, "Environmental Sensor Networks: A revolution in the earth system science?," *Earth-Science Reviews*, vol. 78, pp. 177–191, July 2006.

[102] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervasive Computing*, vol. 6, pp. 50–57, April 2007.

[103] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *ACM SIGPLAN Notices*, vol. 37, pp. 96–107, October 2002.

[104] T. Naumowicz, R. Freeman, A. Heil, M. Calsyn, E. Hellmich, A. Brändle, T. Guilford, and J. Schiller, "Autonomous monitoring of vulnerable habitats using a wireless sensor network," in *Proceedings of the Workshop on Real-World Wireless Sensor Networks*, pp. 51–55, ACM, March 2008.

[105] D. Anthony, W. P. Bennett, M. C. Vuran, M. B. Dwyer, S. Elbaum, A. Lacy, M. Engels, and W. Wehtje, "Sensing Through the Continent: Towards Monitoring Migratory Birds Using Cellular Sensor Networks," in *Proceedings of the ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, pp. 329–340, 2012.

[106] R. N. Handcock, D. L. Swain, G. J. Bishop-Hurley, K. P. Patison, T. Wark, P. Valencia, P. Corke, and C. J. O'Neill, "Monitoring Animal Behaviour and Environmental Interactions Using Wireless Sensor Networks, GPS Collars and Satellite Remote Sensing," *Sensors*, vol. 9, pp. 3586–3603, May 2009.

[107] G. Z. Yang, *Body Sensor Networks*. Springer-Verlag New York, Inc., 2006.

[108] K. H. Farmer, A. Jamart, and B. Goossens, "The re-introduction of chimpanzees Pan troglodytes troglodytes to the Conkouati-Douli National Park, Republic of Congo," in *Soorae, P. S. (ed.) Global re-introduction perspectives: additional case-studies from around the globe: IUCN/SSC Re-introduction Specialist Group*, pp. 231–237, 2010.

[109] R. Kays, B. Kranstauber, P. Jansen, C. Carbone, M. Rowcliffe, T. Fountain, and S. Tilak, "Camera traps as sensor networks for monitoring animal communities," in *IEEE Conference on Local Computer Networks (LCN)*, pp. 811–818, IEEE, 2009.

[110] C. Walzer and G. Fluch, "Small VHF implants for radio-tracking reintroduced, free-rangimg oranguatans (Pongo pygmaeus)," in *Symposium of the International Society on Biotelemetry*, pp. 25–25, May 2012.

[111] H. R. Trayford and K. H. Farmer, "An assessment of the use of telemetry for primate reintroductions," *Journal for Nature Conservation*, vol. 20, no. 6, pp. 311–325, 2012.

[112] D. Fossey, "Observations on the home range of one group of mountain gorilla (gorilla gorilla beringei)," *Animal Behaviour*, vol. 22, pp. 568–581, 1974.

[113] M. M. Robbins and A. McNeilage, "Home Range and Frugivory Patterns of Mountain Gorillas in Bwindi Impenetrable National Park, Uganda," *International Journal of Primatology*, vol. 24, no. 3, pp. 467–491, 2003.

[114] M. Boos, M. Kolbe, P. Kappeler, and E. T., *Coordination in Human and Primate Groups*. Springer-Verlag New York, Inc., 2011.

[115] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM, pp. 53–60, August 1999.

[116] B. Zhou, K. Xu, and M. Gerla, "Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pp. 223–230, November 2004.

[117] M. Musolesi, S. Hailes, and C. Mascolo, "An ad hoc mobility model founded on social network theory," in *Proceedings of MSWiM*, pp. 20–24, October 2004.

[118] M. D. d. A. V. Borrel and S. Fdida, "A preferential attachment gathering mobility model," *IEEE Communications Letters*, vol. 9, pp. 900–902, October 2005.

[119] A. L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, pp. 509–512, October 1999.

[120] C. Sueur, A. Jacobs, F. Amblard, O. Petit, and A. J. King, "How can social network analysis improve the study of primate behavior?," *American Journal of Primatology*, vol. 73, no. 8, pp. 703–719, 2011.

[121] G. Schino, "Grooming, competition and social rank among female primates: a meta-analysis," *Animal Behaviour*, vol. 62, no. 2, pp. 265–271, 2001.

[122] F. De Waal, "Rank distance as a central feature of rhesus monkey social organization: a sociometric analysis," *Animal Behaviour*, vol. 41, no. 3, pp. 383–395, 1991.

[123] L. J. Brent, J. Lehmann, and G. Ramos-Fernández, "Social network analysis in the study of nonhuman primates: A historical perspective," *American Journal of Primatology*, vol. 73, no. 8, pp. 720–730, 2011.

[124] C. Kasper and B. Voelkl, "A social network analysis of primate groups," *Primates*, vol. 50, no. 4, pp. 343–356, 2009.

[125] F. E. Clark, "Space to choose: network analysis of social preferences in a captive chimpanzee community, and implications for management," *American Journal of Primatology*, vol. 73, no. 8, pp. 748–757, 2011.

[126] C. Sueur and O. Petit, "Organization of group members at departure is driven by social structure in Macaca," *International Journal of Primatology*, vol. 29, no. 4, pp. 1085–1098, 2008.

[127] I. Matsuda, P. Zhang, L. Swedell, U. Mori, A. Tuuga, H. Bernard, and C. Sueur, "Comparisons of intraunit relationships in nonhuman primates living in multilevel social systems," *International Journal of Primatology*, vol. 33, no. 5, pp. 1038–1053, 2012.

[128] J. C. Flack, M. Girvan, F. B. De Waal, and D. C. Krakauer, "Policing stabilizes construction of social niches in primates," *Nature*, vol. 439, no. 7075, pp. 426–429, 2006.

[129] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 346–357, Sept. 2006.

[130] V. Ravelomana, "Extremal properties of three-dimensional sensor networks with applications," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 246–257, Jul/Sep 2004.

[131] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks," in *Proceedings of International Conference on UnderWater Networks and Systems (WUWNet)*, pp. 48–55, Sept. 2006.

[132] H. Zhou, S. Xia, M. Jin, and H. Wu, "Localized Algorithm for Precise Boundary Detection in 3D Wireless Networks," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, pp. 744–753, June 2010.

[133] X. Bai, C. Zhang, D. Xuan, and W. Jia, "Full-Coverage and k-Connectivity (k =14, 6) Three Dimensional Networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 388–396, April 2009.

[134] P. I.-S. Chiang and W.-C. Peng, "Slab Routing: Adapting Two-Dimensional Geographic Routing to Three-Dimensions," in *Proceedings of the IEEE SECON*, pp. 1–9, July 2009.

[135] G. Astuti, G. Giudice, D. Longo, C. D. Melita, G. Muscato, and A. Orlando, "An Overview of the 'Volcan Project': An UAS for Exploration of Volcanic Environments," *J. Intell. Robotic Syst.*, vol. 54, pp. 471–494, March 2009.

[136] A. Purohit and P. Zhang, "Sensorfly: A controlled-mobile aerial sensor network," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 327–328, November 2009.

[137] J. Elston and E. W. Frew, "Hierarchical distributed control for search and tracking by heterogeneous aerial robot networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 170–175, May 2008.

[138] J. Elston, E. W. Frew, D. Lawrence, P. Gray, and B. Argrow, "Net-centric communication and control for a heterogeneous unmanned aircraft system," *J. Intell. Robotics Syst.*, vol. 56, pp. 199–232, September 2009.

[139] M. Dumiak, "Robocopters unite!," *IEEE Spectrum*, vol. 46, p. 12, February 2009.

[140] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "SensorFlock: An airborne wireless sensor network of micro-air vehicles," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 117–129, November 2007.

[141] P. Li, C. Zhang, and Y. Fang, "The capacity of wireless ad hoc networks using directional antennas," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1374–1387, October 2011.

[142] J. Jiang, C. Lin, and Y. Hsu, "Localization with Rotatable Directional Antennas for Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Parallel Processing Workshops (ICPPW)*, pp. 542–548, Sept. 2010.

[143] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das, "MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access," in *Proceedings of the International Conference on Mobile Systems, Applications and Services (MobiSys)*, pp. 192–205, June 2007.

[144] G. Jakllari, W. Luo, and S. V. Krishnamurthy, "An Integrated Neighbor Discovery and MAC Protocol for Ad Hoc Networks Using Directional Antennas," *IEEE Transactions on Wireless Communications*, vol. 6, no. 3, pp. 1114–1024, 2007.

[145] A. Alshbatat and L. Dong, "Performance analysis of mobile ad hoc unmanned aerial vehicle communication networks with directional antennas," *International Journal of Aerospace Engineering*, vol. 2010, 2011.

[146] K.-P. Lin and K.-C. Hung, "An efficient fuzzy weighted average algorithm for the military UAV selecting under group decision-making," *Knowledge-Based Systems*, vol. 24, no. 6, pp. 877–889, 2011.

[147] J. Y. Chen, "UAV-guided navigation for ground robot tele-operation in a military reconnaissance environment," *Ergonomics*, vol. 53, no. 8, pp. 940–950, 2010.

[148] M. Israel, "A UAV-based roe deer fawn detection system," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, p. 1, 2011.

[149] S. Sasa, Y. Matsuda, M. Nakadate, and K. Ishikawa, "Ongoing research on disaster monitoring UAV at JAXAs Aviation Program Group," in *Proceedings of the SICE Annual Conference*, pp. 978–981, IEEE, August 2008.

[150] G. Tuna, T. V. Mumcu, K. Gulez, V. C. Gungor, and H. Erturk, "Unmanned Aerial Vehicle-Aided Wireless Sensor Network Deployment System for Post-disaster Monitoring," in *Emerging Intelligent Computing Technology and Applications*, pp. 298–305, Springer, 2012.

[151] S. M. Adams and C. J. Friedland, "A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management," in *Proceedings of the International Workshop on Remote Sensing for Disaster Response*, pp. 15–16, September 2011.

[152] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small fixed-wing unmanned air vehicles: A survey," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 3144–3149, 2007.

[153] M. E. Dempsey, "Eyes of the army–US Army roadmap for unmanned aircraft systems 2010–2035," *US Army UAS Center of Excellence, Ft. Rucker, Alabma*, vol. 9, 2010.

[154] N. V. Sidgwick and H. M. Powell, "Bakerian Lecture. Stereochemical Types and Valency Groups," *Proc. R. Soc. A*, vol. 176, pp. 153–180, October 1940.

[155] R. Gillespie and R. Nyholm, "Inorganic stereochemistry," *Quart. Rev. Chem. Soc.*, vol. 11, pp. 339–380, 1957.

[156] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings, "Bracketology: How can math help? Mathematics and Sports," *Dolcani Mathematical Expositions, Joseph Gallian, ed. Mathematical Association of America*, vol. 43, pp. 55–70, 2010.

[157] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 33–46, February 2003.

[158] "OPNET Modeler." http://www.opnet.com.

[159] N. Burri, P. von Rickenbach, R. Wattenhofer, and Y. Weber, "Topology Control Made Practical: Increasing the Performance of Source Routing," *Lecture Notes in Computer Science*, vol. 4325, pp. 1–12, December 2006.

[160] K. S. Trivedi, *Probability and statistics with reliability, queuing and computer science applications*. Chichester, UK: John Wiley and Sons Ltd., 2001.

[161] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A tool for simple connectivity assessment in lossy environments," *Technical Report, Center for Embedded Network Sensing (CENS)-UCLA*, June 2003.

[162] J. M. A. Peixoto, B. W. Nelson, and F. Wittmann, "Spatial and temporal dynamics of river channel migration and vegetation in central Amazonian white-water floodplains by remote-sensing techniques," *Remote Sensing of Environment*, vol. 113, no. 10, pp. 2258 – 2266, 2009.

[163] A. V. M. Cencini, G. Lacorata and E. Zambianchi, "Mixing in a meandering jet: a markovian approximation," *J. Phys. Ocean.*, vol. 29, no. 10, pp. 2578–2594, 1999.

[164] A. S. Bower, "A simple kinematic mechanism for mixing fluid parcels across a meandering jet," *J. Phys. Ocean.*, vol. 21, no. 1, pp. 173–180, 1991.

[165] R. M. Samelson, "Fluid exchange across a meandering jet," *J. Phys. Oceanogr.*, vol. 22, pp. 431–444, 1992.

[166] M. F. Shlesinger, J. Klafter, and Y. Wong, "Random Walks with Infinite Spatial and Temporal Moments," *Journal of Statistical Physics*, vol. 27, pp. 499–512, March 1982.

[167] W. W. Denham, “Energy Relations and Some Basic Properties of Primate Social Organization1,” *American Anthropologist*, vol. 73, no. 1, pp. 77–95, 1971.

[168] G. Caldarelli, “Scale-free networks: complex webs in nature and technology,” *OUP Catalogue*, 2007.

[169] R. P. D. Atkinson, C. J. Rhodes, D. W. Macdonald, and R. M. Anderson, “Scale-free dynamics in the movement patterns of jackals,” *OIKOS*, vol. 98, pp. 134–140, July 2002.

[170] J. Ramos-Fernandez, L. Morales, O. Miramontes, G. Cocho, H. Larralde, and B. Ayala-Orozco, “Lévy walk patterns in the foraging movements of spider monkeys (ateles geoffroyi),” *Behavioural Ecology and Sociobiology*, vol. 55, pp. 223–230, 2004.

[171] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. da Luz, E. P. Raposo, and H. E. Stanley, “Optimizing the success of random searches,” *Nature*, vol. 401, pp. 911–914, October 1999.

[172] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, “On the Lévy-Walk Nature of Human Mobility,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 924–932, April 2008.

[173] D. J. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton university press, 1999.

[174] R. Dunbar, *How Many Friends Does One Person Need?: Dunbar's Number and Other Evolutionary Quirks*. Faber and Faber, February 2010.

[175] B. Okdie, R. Guadagno, F. Bernieri, A. Geers, and A. Mclarney-Vesotski, “Getting to know you: Face-to-face versus online interactions,” *Computers in Human Behavior*, vol. 27, pp. 153–159, Jan. 2011.

[176] C. M. K. Debra L. Oswald, Eddie M. Clark, “Friendship Maintenance: An Analysis of Individual and Dyad Behaviors,” *Journal of Social and Clinical Psychology*, vol. 23, pp. 413–441, October 2004.

[177] A. Sutcliffe and D. Wang, “Computational Modelling of Trust and Social Relationships,” *Journal of Artificial Societies and Social Simulation*, vol. 15, p. 3, January 2012.

[178] M. Butovskaya, A. Kozintsev, and C. Welker, “Grooming and social rank by birth: the case of Macaca fascicularis,” *Folia Primatologica*, vol. 65, no. 1, pp. 30–33, 1995.

[179] T. G. O'Brien, “Allogrooming behaviour among adult female wedge-capped capuchin monkeys,” *Animal Behaviour*, vol. 46, no. 3, pp. 499–510, 1993.

[180] J. Clune, J.-B. Mouret, and H. Lipson, "The evolutionary origins of modularity," *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1755, 2013.

[181] J. Banford and J. Irvine, "Estimating Social Graphs in an Education Environment," *IEEE Vehicular Technology Magazine*, vol. 7, pp. 31–37, Mar. 2012.

[182] E. M. Boucher, J. T. Hancock, and P. J. Dunham, "Interpersonal Sensitivity in Computer-Mediated and Face-to-Face Conversations," *Media Psychology*, vol. 11, pp. 235–258, June 2008.

[183] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal Ubiquitous Computing*, vol. 10, pp. 255–268, March 2006.

[184] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying Ad-Hoc Networks (FANET): A Survey," *Ad Hoc Networks*, vol. 11, pp. 1254–1270, May 2013.

[185] L. Kesheng, Z. Jun, and Z. Tao, "The clustering algorithm of UAV Networking in Near-space," in *International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, pp. 1550–1553, 2008.

[186] C. Konstantopoulos, D. Gavalas, and G. Pantziou, "A mobility aware technique for clustering on mobile ad-hoc networks," in *Distributed Computing and Networking*, vol. 4308 of *Lecture Notes in Computer Science*, pp. 397–408, 2006.

[187] S. Thrun, M. Diel, and D. Hähnel, "Scan Alignment and 3-D Surface Modeling with a Helicopter Platform," *Field and Service Robotics*, pp. 287–297, 2006.

[188] V. G. S. Box, "The Molecular Mechanics of Quantized Valence Bonds," *Molecular modeling annual*, vol. 3, no. 3, pp. 124–141, 1997.

[189] K. Li, N. Ahmed, S. S. Kanhere, and S. Jha, "Reliable communications in aerial sensor networks by using a hybrid antenna," in *Proceedings of the IEEE Local Computer Networks (LCN)*, pp. 156–159, October 2012.

[190] N. Ahmed, S. S. Kanhere, and S. Jha, "Link Characterization for Aerial Wireless Sensor Networks," in *Proceedings of the IEEE Global Communicatios Conference (GLOBECOM) Workshops*, pp. 1274–1279, December 2011.

[191] R. J. Gillespie, "The electron-pair repulsion model for molecular geometry," *Journal of Chemical Education*, vol. 47, p. 18, January 1970.

[192] S. F. A. Kettle *Theoret. Chim. Acta*, vol. 3, p. 211, 1965.

[193] J. Clapper, J. Young, J. Cartwright, and J. Grimes, "Unmanned Systems Roadmap 2007-2032," tech. rep., Dept. of Defense, 2007.

[194] G. Lee, Y. Nishimura, K. Tatara, and N. Y. Chong, "Three dimensional deployment of robot swarms," in *Proceedings of the IEEE/RSJ IROS*, pp. 5073–5078, October 2010.

[195] Z. Alliance, "Zigbee specification," *Document 053474r06, Version*, vol. 1, 2006.

[196] "UAV flight time," April 2012.