Distributed Localization and Conflict Detection in Mobile Wireless Networks

Imen Mahjri

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Cotutelle internationale *Ecole Nationale des Sciences de l'Informatique*

Présentée et soutenue le *29/09/2017* **par :**

IMEN MAHJRI

Distributed Localization and Conflict Detection in Mobile Wireless Networks

---

JURY

ALFREDO GRIECO	Professeur Associé	Président du Jury
FRANCINE KRIEF	Professeur	Rapporteur
MAHER BENJEMAA	Professeur	Rapporteur
IMEN JEMILI	Maître de Conférences	Examinateur
KHALIL DRIRA	Directeur de Recherche	Directeur de thèse
ABDELFETTAH BELGHITH	Professeur	Directeur de thèse
SAMIR MEDJIAH	Maître de Conférences	CoDirecteur de thèse
AMINE DHRAIEF	Maître de Conférences	CoDirecteur de thèse

École doctorale et spécialité :

MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture

Unité de Recherche :

Laboratoire d'Analyse et d'Architecture des Systémes (LAAS-CNRS)

Directeur(s) de Thèse :

Khalil DRIRA et *Abdelfettah BELGHITH*

Rapporteurs :

Francine KRIEF et *Maher BENJEMAA*

Abstract

A mobile wireless network is a collection of mobile nodes connected via wireless links. The mobile node can be a person carrying a mobile terminal, an animal with an embedded sensor (electronic tag), a vehicle or any other mobile device with wireless communication capabilities. This thesis deals with two fundamental issues in mobile wireless networks: localization and conflict detection. Localization is the estimation of the absolute or relative positions of the mobile nodes. Conflict detection is the prediction of potential future conflicts between the mobile nodes. A conflict is a situation in which two or more mobile nodes are within an unsafe distance from one another. Conflict detection is a crucial requirement for mobile networks where the nodes can get too close to each other and collide.

The first contribution of this thesis is a new weighted localization algorithm for mobile wireless networks. The proposed algorithm is distributed and requires low computational and communication overheads enabling its use in resource-limited nodes. In particular, a small set of nodes with known positions, called beacon nodes, are periodically broadcasting their coordinates. A mobile node with an unknown position collects the beacon nodes coordinates, attributes them weights based on different metrics (e.g, link quality, reception time) and finally estimates its position as the weighted average of all the collected coordinates. The second contribution of this thesis consists in a straight line conflict detection algorithm called SLIDE. SLIDE is distributed and lightweight: each mobile node uses simple mathematical inequalities in order to accurately and timely predict future conflicts with the surrounding nodes. Furthermore, SLIDE drops the restrictive assumption of perfect sensing capabilities and perturbation-free environment in order to guarantee its efficiency in real world applications. The third and last contribution of this thesis is a stochastic model that assesses the conflicts risks in a mobile wireless network where the mobile nodes are moving in the same shared space. Unlike most of the existing stochastic models, the proposed model is simple, generic and requires only two input parameters: the number of mobile nodes and the parameter λ characterizing the inter-contact time between a pair of mobile nodes. The parameter λ may be difficult to estimate experimentally. We therefore provide a generic explicit expression for λ and then specify this generic expression for two commonly used mobility models. All the contributions of this thesis are validated through extensive simulations based on the discrete-event simulator OMNeT++.

Keywords: Mobile wireless networks, distributed localization, distributed conflict detection.

Résumé

Un réseau sans fil mobile est une collection de noeuds mobiles connectés via des liens sans fil. Le noeud mobile peut être une personne portant un terminal mobile, un animal avec une étiquette électronique, un véhicule ou tout autre appareil mobile doté de capacités de communication sans fil. Cette thèse traite deux importants problèmes dans les réseaux sans fil mobiles : la localisation et la détection des conflits. La localisation est l'estimation des positions absolues ou relatives des noeuds mobiles. La détection des conflits est la prédiction des potentiels futurs conflits entre les noeuds mobiles. Un conflit est une situation dans laquelle deux ou plusieurs noeuds mobiles se trouvent à une distance inférieure à une distance minimale requise. La détection des conflits est particulièrement nécessaire dans les réseaux mobiles où les noeuds risquent de trop se rapprocher les uns des autres et se heurter.

La première contribution de cette thèse est un nouvel algorithme de localisation pour les réseaux sans fil mobiles. L'algorithme proposé est distribué et nécessite un faible coût de calcul et de communication permettant son utilisation par des noeuds à ressources limitées. En particulier, un petit ensemble de noeuds avec des positions connues, appelés des noeuds ancres, diffusent périodiquement leurs coordonnées. Chaque noeud mobile avec une position inconnue collecte les coordonnées des noeuds ancres, les attribue des poids en fonction de différentes métriques (par exemple, la qualité de la liaison sans fil, le temps de réception) et enfin estime sa position en tant que la moyenne pondérée de toutes les coordonnées collectées. La deuxième contribution de cette thèse consiste en un algorithme de détection de conflit nommé SLIDE. SLIDE est distribué et à faible complexité: chaque noeuds mobile utilise des simples inégalités mathématiques afin de prédire avec précision et en temps opportun les futurs conflits avec les noeuds voisins. En outre, SLIDE abandonne l'hypothèse restrictive des positions et vitesses précises ainsi que des communications non perturbées afin de garantir son efficacité dans les applications du monde réel. La troisième et dernière contribution de cette thèse est un modèle stochastique qui évalue les risques des conflits dans un réseau sans fil mobile où les noeuds mobiles se déplacent dans le même espace partagé. Contrairement à la plupart des modèles stochastiques existants, le modèle proposé est simple, générique et ne nécessite que deux paramètres d'entrée: le nombre de noeuds mobiles et le paramètre λ caractérisant le temps d'inter-contact entre une paire de noeuds mobiles. Le paramètre λ peut être difficile à estimer expérimentalement. Nous proposons donc une expression générique pour λ et puis nous la spécifions pour deux modèles de mobilité couramment utilisés. Toutes les contributions de cette thèse

sont validées en utilisant des simulations basées sur le simulateur d'événements discrets OMNeT ++.

Mots-clés: Réseaux sans fil mobiles , localisation distribuée, détection distribuée des conflits.

Contents

List of Figures

List of Tables

Introduction

Mobile wireless networks have attracted lots of research attention due to their wide use in many commercial and public service applications such as medical care, intelligent transportation, home automation and industrial monitoring. A mobile wireless network consists of a set of mobile nodes that can communicate over a wireless medium. A mobile node may be an individual carrying a mobile phone, an animal with an implanted tag, a vehicle or any other mobile device with wireless communication capabilities. The communication of the mobile nodes includes: the communication between a mobile node and a fixed device (e.g., an access point) and/or the communication between mobile nodes.

In this thesis, we treat two fundamental problems in mobile wireless networks which are localization and conflict detection. Location awareness is an essential feature for many applications of mobile wireless networks. Indeed, the information collected or communicated by the wireless mobile nodes is usually valueless without the knowledge of the nodes location. Location information also enhances the interaction between the nodes and their surroundings. Mobile wireless nodes could be equipped with a global positioning system (GPS) to obtain their locations, but this is currently a costly solution (energy consumption, production price, size of the node). Besides, GPS service may be inaccessible in some environments such as mountains, dense forests and indoors. Thus, an alternative solution to GPS is strongly required. The first contribution of this thesis consists in a new weighted localization algorithm for mobile wireless networks called ASAW. In the proposed algorithm a small number of nodes, referred to as anchors, are aware of their own positions by either using GPS or being manually configured. The anchor nodes may be static or mobile. The mobile nodes that are unaware of their locations (called unknowns), collect the location information from their neighbouring location aware nodes, weight them based on different metrics (e.g, link quality, distance, velocity, etc) and finally estimate their positions as the weighted average of the coordinates of the collected location information. The proposed algorithm have low computational and communication costs which enables its implementation in resource-constrained nodes.

The second research problem treated in this thesis is conflict detection in mobile wireless networks. A conflict is a situation in which two or more mobile nodes experience a

loss of minimum separation. That is, a conflict occurs when the distance between two or more mobile nodes is less than a minimum required separation distance. This separation requirement prevents nodes from getting too close from each other and colliding. The beforehand detection of conflicts is necessary for the avoidance of collisions between nodes and hence the loss of the mobile nodes. In this context, the second contribution consists in a new spatio-temporal conflict detection and alerting algorithm for mobile communicating nodes. The proposed algorithm called SLIDE is fully distributed and requires a limited state information exchange between the mobile nodes in order to detect future conflicts. In contrast to previous approaches, we alleviate the strong requirement for precise state information and packet-loss free communications so as to guarantee the applicability and efficiency of the algorithm in real world situations. Extensive simulations based on the OMNeT++ simulator are used to validate SLIDE and evaluate its performance. Simulation results indicate that SLIDE guarantees a reduced number of false and missed alarms even in high density traffic scenarios and communication perturbed environment, yet it leaves adequate time to accomplish the avoidance actions.

The third and final contribution of this thesis is a stochastic model that accurately models conflicts in a mobile wireless network where nodes are moving in the same shared space. The model has only two input parameters, namely the number of mobile nodes and the parameter of an exponential distribution which describes the time that it takes for an arbitrary mobile node to come into contact with another mobile node. Using only these two parameters, we provide simple, yet accurate closed-form expressions for different conflict related metrics such as safety periods, survival probabilities and number of conflicts. Two scenarios are particularly considered. The first scenario assumes that mobile nodes are equipped with perfect conflict detection and avoidance capabilities. The aim of this first scenario is to answer the question of whether a detection and resolution equipment is required for a given mobile nodes swarm characterized by its number of nodes N and its parameter λ. The second scenario assumes that the mobile nodes are not equipped with conflict detection and avoidance capabilities. This can especially be the case for small mobile nodes that can not satisfy the requirements of conflict detection and avoidance equipage because of size, weight or power constraints. In addition, a generic explicit expression is given for the parameter λ. This generic expression is then specified for two mobility models. Extensive simulations based on the OMNeT++ simulator are used to validate the obtained analytical results. The simulation results are shown to be in close agreement with the analytical results.

This thesis is organized as follows. Chapter 1 presents an overview and a taxonomy of the existing localization techniques in wireless networks. Chapter 2 reviews the state of the art and the foundations of conflict detection and avoidance in mobile wireless networks.

Chapter 3 details our first contribution consisting in a new weighted localization algorithm for mobile wireless networks. Chapter 4 introduces SLIDE our novel Straight LIne conflict DEtection and alerting algorithm for 3D-mobile communicating nodes. Chapter 5 details our third contribution; a stochastic model evaluating the conflict risks in a swarm of 3D-mobile nodes sharing the same airspace. Chapter 6 concludes the thesis by summarizing the major contributions and providing ideas and directions for future work.

Chapter 1

Localization in Wireless Networks: State of the Art and Foundations

Contents

Localization of wireless devices is a crucial requirement for many emerging applications such as environmental monitoring, intelligent transportation, home automation, healthcare monitoring and social networking. For instance, in an environmental monitoring application such as forest fire detection or air pollution monitoring, the collected information is worthless without the location of nodes. Wireless nodes could be equipped with a GPS to acquire their locations, but this is currently a costly solution in terms of energy and price. Thus, in the recent years, several localization algorithms that aim at obtaining nodes locations with a lower cost have been proposed. In this chapter we give a review of the state of the art research concerning localization in wireless networks. We first

present the localization problem formulation, we then propose a taxonomy of the existing localization techniques and finally detail some representative localization algorithms.

1.1 Definitions and Problem Formulation

Some definitions are first needed in order to understand the localization problem.

Definition 1 (Unknown Nodes ($\mathcal{U}$)). *A node $u \in \mathcal{U} \iff u$ is not aware of its own position. Unknown nodes are also referred to as normal nodes or blindfolded nodes.*

Definition 2 (Anchor nodes ($\mathcal{A}$)). *A node $a \in \mathcal{A} \iff a$ is aware of its own position (through manual configuration or GPS). Anchor nodes are also called beacon nodes, landmarks or reference nodes.*

Definition 3 (Localization problem). *Given a network with a set $\mathcal{N}$ of nodes, m anchor nodes in $\mathcal{A} \subset \mathcal{N}$ with known positions $\{X_1, ..., X_m\}$, k unknown nodes in $\mathcal{U} \subset \mathcal{N}$ with unknown positions $\{X_{m+1}, ..., X_{m+k}\}$, estimate the positions $\{\widehat{X}_{m+1}, ..., \widehat{X}_{m+k}\}$ of the unknown nodes as close as possible to their true positions $\{X_{m+1}, ..., X_{m+k}\}$.*

1.2 Taxonomy of Localization Techniques

We provide in this section a taxonomy of the existing localization techniques. This taxonomy provides general guidelines for understanding the differences between the existing localization techniques.

1.2.1 Target vs. self-localization

Depending on their final goal and on their different fields of application, localization techniques can be categorized into two groups: target localization and self-localization.

The objective of target localization is to determine the location of a target (e.g., human, animal, vehicle, device). Target localization can be classified into two categories: active target localization and passive target localization. In active target localization the target actively emits a specific signal that can be received and analyzed by a reader [Savi 16]. Active target localization has a broad range of applications such as asset inventory and resource management. In passive target localization, the target does not actively participate in the localization process, it is rather just a reflecting/scattering object [Han 14]. Passive target localization is crucial for many applications such as crimes prevention and tracking, surveillance and medical patient monitoring.

In self-localization, unknown nodes determine their positions by themselves. Self-localization can be classified into two categories: active self-localization and passive self-localization. In passive self-localization, existing beacon signals are used by the unknown nodes to passively deduce their own positions [Hadd 16]. In active self-localization, unknown nodes actively inquire the location information from their surrounding environment [Reza 11]. Self-localization is necessary in many applications such as environmental monitoring applications where the measurement data are worthless without the location of the measuring node.

1.2.2 Centralized vs. distributed localization

Centralized localization algorithms are designed to run on a sufficiently powerful central base station [Tomi 15]. First the base station collects the environmental information from the different sensor nodes. Then, based on the collected information, it computes the position of each sensor node and migrates them back to the respective nodes. Centralized algorithms eliminate the problem of nodes computational limitations but they introduce a large communication cost due to transporting data to and from the base station. Hence, centralized algorithms are only suitable for small networks. In contrast, distributed algorithms are designed to run on each node [Meye 16]. Unknown nodes positions are estimated based only on the inter-nodes communication. Due to the lack of global information, distributed localization is usually less accurate than the centralized one but it considerably reduces the communication costs. Figure 1.1 illustrates the difference between the centralized and the distributed techniques.

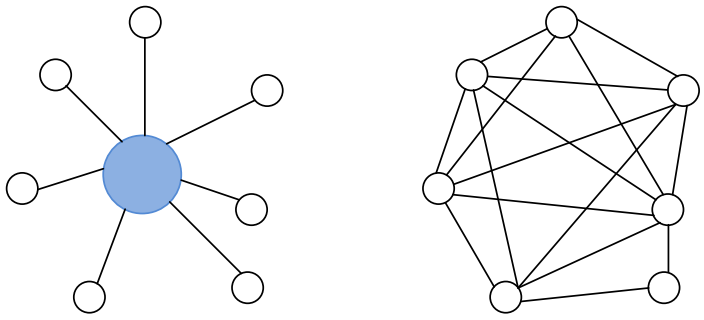


Figure 1.1 – Centralized vs. distributed localization techniques

1.2.3 Range-based vs. range-free localization

Depending on their ranging assumption, localization techniques can be divided into range-based and range-free.

Range-based localization

Range-based algorithms use inter-nodes distances or angles to estimate the nodes positions. They use special *measurements techniques* such as the time of arrival, the angle of arrival, and the received strength of a given transmitted *signal* to calculate the distance or angle separating two sensors.

Signal technologies: the choice of the signal technology used by sensor nodes for localization depends on the considered environment, application as well as the required precision, range and cost. These technologies include infrared (IR) [Seke 15], ultrasound [Filo 10], magnetic [Song 13], optical [Suh 16] and radio frequency signals. Radio technology is the most widely used technology for localization. Depending on the type of the used frequency range, the radio frequency signals can be classified into different groups: radio frequency identification (RFID) [DiGi 14], WIFI (IEEE 802.11) [Yang 15], zigbee (IEEE 802.15.4) [Chen 11], bluetooth (IEEE 802.15.1) [Gu 15], wide area cellular [Abu 16] and ultra-wideband (UWB) [Reyn 13]. Table 1.1 summarizes the different signal technologies used for localization.

Measurement techniques: There are three major measurement techniques to determine the distance/angel between the nodes: the Time Of Arrival (TOA), the Angle Of Arrival (AOA) and the Received Signal Strength (RSS). In the TOA measurement technique, the distance separating a receiver from a sender is calculated through multiplying the propagation time by the speed of the signal [Shen 12]. TOA based techniques require a direct line-of-sight path between the transmitter and the receiver. The presence of obstacles in between them leads to later-arriving signals and hence inaccurate ranging estimations. Time synchronization between the transmitter and the receiver is also usually needed. There are however some existing works where TOA measurements are done without time synchronization [Chen 12a]. The AOA measurement technique typically relays on the use of radio or microphone arrays to estimate the angel separating the receiver from the transmitter [Wang 15]. Systems based on the AOA measurement technique require specific hardware, they are thus expensive in terms of manufacturing cost, energy consumption and complexity. The accuracy of AOA based techniques also degrades as the distance between the transmitter and the measuring unit increases. The RSS measurement technique depends on the fact that the signal strength attenuates with distance [Yagh 14]. With the attenuation information, a receiving node is able to calcu-

Technology	Remarks
Infrared	• Widely used in indoor localization • Low cost and low power • Requires close proximity and line of sight between the transmitter and the receiver • Is difficult to read in the presence of sunlight • Typical range: up to 5m
Ultrasound	• Cannot penetrate through the walls • Affected by reflected signals and other noise sources (e.g., jangling metal objects) • Typical range: 3-10m
Magnetic	• Magnetic sensors are small and cheap • No line of sight requirement • Typical range: 1-3m
Optical	• Requires line of sight • Affected by many interference sources (e.g., light, weather) • Typical range: up to 5m
Radio frequency	• Can pass through buildings, human body and other obstructions • Affected by multipath 1) RFID: – Light and small tags that can be attached to people or equipments – Typical range: 1-10m 2) WIFI: – Uses the existing WLAN infrastructure for localization: lower cost – Typical range: 50-100m 3) Zigbee: – Low cost and low power – Typical range: 10-30m 4) Bluetooth: – Low cost and low power – Typical range: 10-15m 5) Cellular – Localizes a mobile within a cell coverage area – Typical range: 100-150m
UWB	• No line of sight requirement • Less multipath distortion than the other RF technologies • High penetration • Typical range: 10m

Table 1.1 – Signal technologies used for localization

late its distance to the transmitting node. The RSS based techniques typically use radio signals. Indeed, the use of radio signals do not require any additional hardware since most of the radio communication devices come with built-in RSS indicator (RSSI) hardware

that directly provides the RSS measurements. Radio waves are nevertheless vulnerable to the environmental dynamics which may affect the accuracy of the distance estimations. Some works [Tomi 16] considered hybrid schemes combining two different measurement techniques in order to ameliorate the range estimations. Figure 1.2 illustrates the different range measurement techniques.

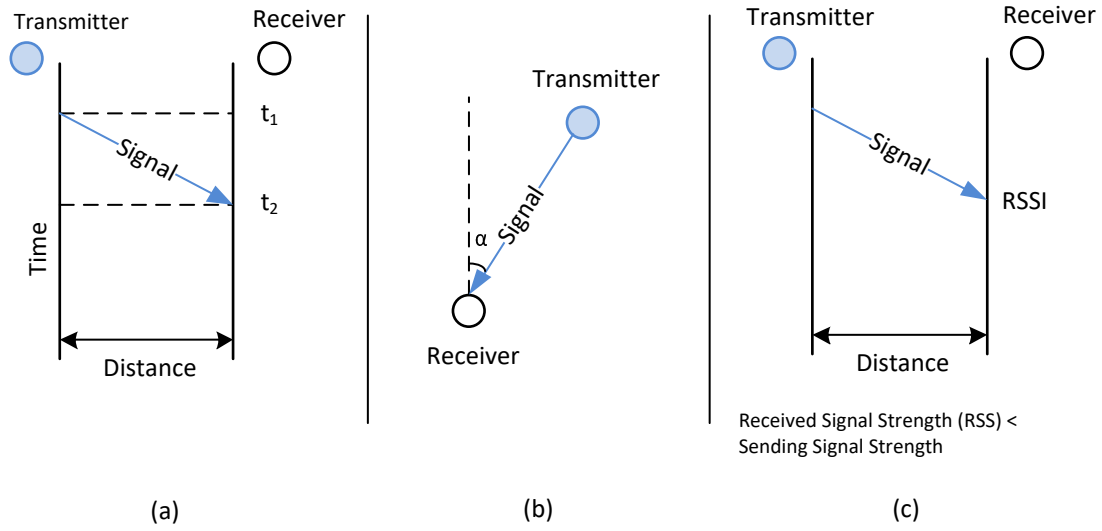


Figure 1.2 – Range based localization: (a) TOA (b) AOA (c) RSS

Range-free localization

Range-free localization algorithms make no assumption about the availability of inter-nodes distances or angles to estimate the locations [Zaid 16]. They instead rely on topology and connectivity information assuming an isotropic network where the hop count between nodes is proportional to their distance. Range-free algorithms provide promising solutions for the localization problem since they do not require extra hardware. However, because of the absence of range information, the positions estimations obtained by these methods are usually less accurate than those obtained by the range-based methods.

1.2.4 Network-based vs. non-network-based localization

The network-based localization techniques use the already existing network infrastructure, such as WLAN, and consequently avoid the expensive and the time-consuming installation of the localization infrastructure [Wu 13]. The non-network-based localization techniques use dedicated infrastructure for positioning, such as sensor-based positioning systems [Suh 16]. The non-network-based localization techniques are more costly and less time

effective than the network-based localization techniques but offer, on the other hand, more control over the physical specifications and hence over the quality of the location estimations.

1.2.5 Outdoor vs. indoor localization

Indoor localization is more challenging than outdoor localization due to the complexity of the indoor environment. The various obstacles (e.g., walls, equipment), the mobility of people and the interference with other networks traffic degrade the accuracy of the positioning. Some works have tried to deal with the complexity of the indoor environment using the fingerprinting technique (also known as scene analysis) [Seet 12, He 16]. In this technique, an offline training phase is used in order to collect the signal features (fingerprints) for a particular indoor scene. The estimated location of a given node is calculated during the online phase based on these offline collected measurements. The drawback of the fingerprinting technique is that it requires a lot of pre-processing work and is ineffective in dynamic and changeable environments. A new training phase should be executed when there is any change in the environment.

1.2.6 Mobile vs. static nodes

Based on the mobility state of the anchor and normal nodes, existing localization algorithms can be classified into four groups: (1) static anchors and static normal nodes (2) static anchors and mobile normal nodes (3) mobile anchors and static normal nodes (4) mobile anchors and mobile normal nodes. The scenario of both static anchors and normal nodes is the most studied and hence the most mature localization scenario. In the second scenario (static anchors and mobile normal nodes), typically a small number of static anchors are mounted in discreet locations like ceilings or walls in order to track or help the unknown nodes estimate their coordinates. In the third scenario (mobile anchors and static normal nodes), a number of mobile anchors traverse the deployment region and periodically broadcast their coordinates. Unknown mobile nodes uses these location announcements in order to infer their own location. In the last scenario (mobile anchors and mobile normal nodes), mobile anchors locations are periodically used in order to localize the set of mobile normal nodes. When the unknown nodes are static (scenarios 1 and 3), the localization process can be executed only once (e.g., during initialization). However, when the unknown nodes are mobile (scenarios 2 and 4), the localization process must be frequently executed in order to determine the continuously changing positions of nodes.

We summarize in Figure 1.3 the different discussed categories and cite representative works in each category.

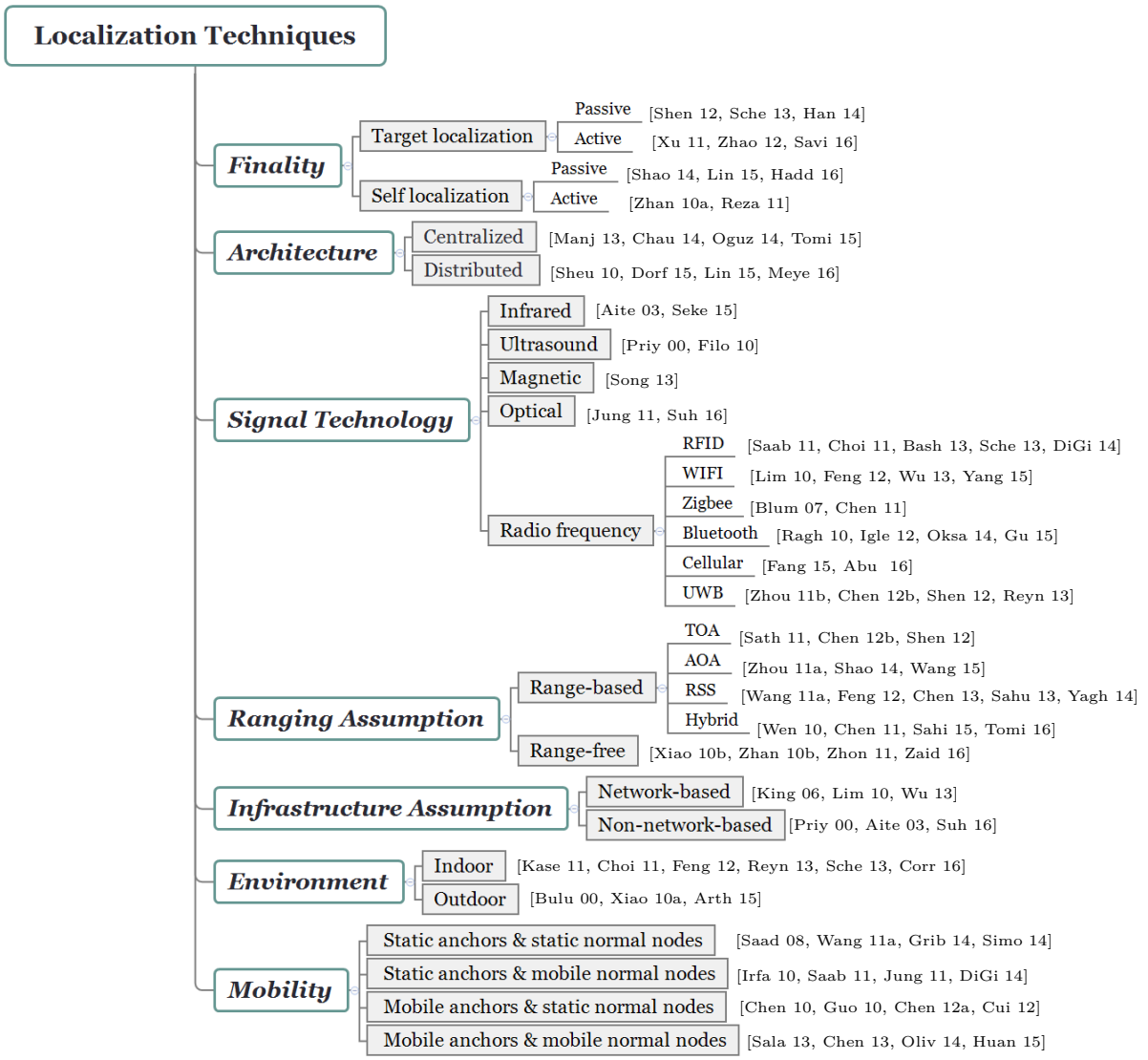


Figure 1.3 – Taxonomy of localization techniques

1.3 Solutions to Localization Problem

We review next a selected set of representative localization algorithms.

Dv-hop: Dv-hop [Nicu 01] is a classic range-free localization algorithm. It works as follows. First all anchor nodes broadcast their locations. The messages are propagated hop by hop to reach all nodes in the network. Each node maintains a table containing all anchors locations and the least number of hops from each anchor. Once an anchor receives the coordinates of all the other anchors, it estimates the average distance per hop and broadcasts it. The average distance per hop d_i an anchor situated at (X_i, Y_i)

computes is calculated as follow:

$$d_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_{ij}},\ i \neq j,\ all\ landmarks\ j \tag{1.1}$$

Where h_{ij} denotes the minimum hop-count between anchors i and j. When receiving the average distance per hop (usually received from the closest anchor), a non-anchor node determines its distance from each anchor by multiplying the least number of hops to the anchor with the average distance per hop. Then it applies a multilateration (positioning using differences in distances) procedure to estimate its location. DV-Hop has the advantage of involving only few beacon nodes. Its considerable disadvantage is that it fails in networks with irregular topologies, where the variance in actual hop distances is very large.

AT-Free: paper [Saad 07] proposes a distributed and range-free localization algorithm called AT-Free. In AT-Free, each anchor broadcasts a message containing its identity, position coordinates and a hop counter parameter initialized to one. When receiving this message, a sensor node increments the hop counter and broadcasts the message. All messages with higher hop counter per anchor are ignored. Thus, each node can obtain the shortest distance, in hops, to all anchors. A non localized node calculates the estimated distance that separates it from an anchor by multiplying the number of hops to the anchor with the communication range R. The estimated position of a non localized node is the center of gravity of the zone defined by the intersection of disks centred in anchors and of radii equal to the estimated distance separating the node to the corresponding anchor. Figure 1.4 illustrates how an unknown node X estimates its position using AT-Free. First, anchor nodes (A, B, C) broadcast messages containing their position coordinates. These messages are propagated hop by hop to reach X. Node X is then able to estimate the number of hops h_a, h_b and h_c from anchors A, B, C respectively. X concludes that it is located within the disks centred at A, B, C and with radii equal to $h_a \times R$, $h_b \times R$, $h_c \times R$ respectively. The intersection of these disks defines the zone Z_X (the hatched zone). X estimates its position as the centroid of this zone.

CDL: [Zhao 13] proposes CDL, a Combined and Differentiated Localization technique that combines both connectivity information and RSSI readings to estimate unknown nodes locations. CDL mainly consists of three phases: the virtual-hop localization, filtration, and calibration. The virtual-hop localization phase initially estimates node locations using connectivity information. First distances to the anchors are calculated using a new metric called virtual-hop-count then the unknown nodes positions are calculated based on trilateration [Bouk 07]. Traditional hop-count-based distance estimation techniques make no difference between two distances having the same hop count. Such approaches

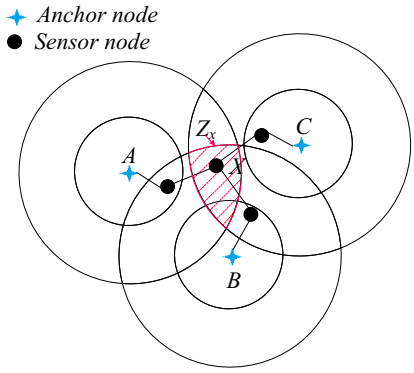


Figure 1.4 – Principle of the localization process in AT-free

may provide inaccurate distance estimations especially in networks with non uniform deployments. The proposed virtual-hop-count based technique makes full use of local information in order to avoid such inaccuracies with non uniform node distributions. During the filtering phase, CDL uses two local filtration techniques, namely the *neighborhood hop-count matching* and the *neighborhood sequence matching*, to identify nodes with good location accuracy. The *neighborhood hop-count matching* technique identifies the good nodes by comparing a node's hop counts to its neighbours. The *neighborhood sequence matching* technique identifies good nodes based on the matching degree between their RSSI sequence and their distance sequence. First each normal node sorts its neighbours in descending order according to the RSSI value from them. This first generated sequence is called RSSI sequence S_1. Second, each node uses the estimated coordinates to calculate the distances to its neighbors and then sorts them in the ascending order according to the calculated distances. This second generated sequence is called the distance sequence S_2. According to the observation that the RSSI monotonically decreases as the distance increases, S_1 and S_2 should be identical. A significant mismatch between S_1 and S_2 indicates an important error in the node's estimated location. In the calibration phase the filtered good nodes are used to calibrate the location of bad ones.

EMAP: paper [Ou 08a] proposes a distributed range-free localization algorithm called EMAP where both anchors and nodes are mobile. It is assumed that all nodes are aware of their accurate moving direction and distance. The proposed algorithm is based on the corollary of perpendicular bisector of a chord stating that *a perpendicular bisector of a chord crosses the center of the circle* (Figure 1.5). During their moving, anchor nodes periodically broadcast beacon packets including their locations. Unknown nodes store the received beacon points and continuously reposition them according to their mobility (distance and direction changes). An unknown node can estimate its location once 3 beacons are received. It constructs two chords which endpoints are the beacons points

and estimates its position as the intersection point of the perpendicular bisectors of these chords. This positioning process is continuously performed during the nodes movement. Once an unknown node receives a new beacon point, it performs the location estimation using two former adjusted beacon points and the newly received one.

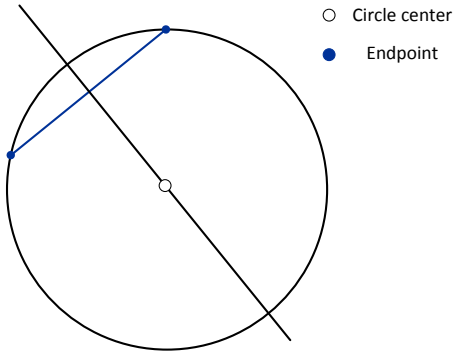


Figure 1.5 – Corollary of perpendicular bisector of a chord

Flying anchor assisted algorithms: the works in [Ou 08b, Kim 10, Fu 11] use flying anchors in order to determine the 3D positions of nodes. In [Ou 08b] a flying beacon with an onboard GPS receiver broadcasts its current location information as it flies through the deployment space. Normal nodes use the location information received from the flying anchor and basic geometry principles to estimate their 3D coordinates. In particular, this scheme is based on the geometric corollary stating that *a perpendicular line that passes through the center of a circular cross section of a sphere also passes through the center of that sphere*. It is assumed that nodes have perfect spherical communication ranges. A normal node collects location information sent by the flying anchor, chooses the appropriate beacon points to form two circular cross sections and then estimates its position as the intersection point of the two perpendicular lines passing through the centers of these two circular cross sections. As shown in Figure 1.6, the position of the unknown node can be obtained by calculating the intersection point I of the lines L_1 and L_2.

AT-Dist: the work in [Saad 08] proposes the Distance Based Approximation Technique AT-Dist. AT-Dist assumes that all sensor nodes have an identical transmission range R and that each node is able to compute its distances to its neighbours when it receives signals. Nodes running AT-Dist first use the distance estimation technique Sum-Dist to estimate distances to anchors and then approximate their positions using this distance information. In Sum-Dist every anchor broadcasts a message including its identity, coordinates, and a path length set to zero. Each receiving node calculates the range from the sender (via RSS or TOA), adds it to the path length and broadcasts the message. Hence, each unknown node in the network can obtain a distance estimation and

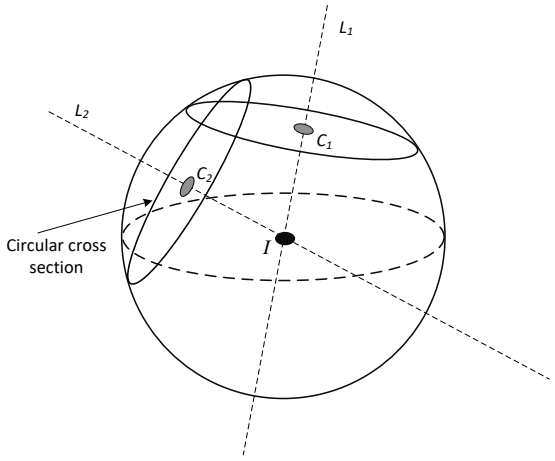


Figure 1.6 – Intersection point of two circular cross sections of a sphere

the position of every anchor. Obviously, only the shortest distance will be conserved. If i_1,i_2,..., i_n are the intermediate nodes from the anchor a to the unknown node u, then the estimated distance $\widehat{d}_{au}$ between a and u is:

$$\widehat{d}_{au} = d_{ai_1} + d_{i_1 i_2} + ... + d_{i_n u}$$

For example, in Figure 1.7 the estimated distance $\widehat{d}_{MN}$ between M and N is $d_{MI} + d_{IN}$. By triangular inequality we have: $d_{MN} \leq \widehat{d}_{MN} = d_{MI} + d_{IN}$

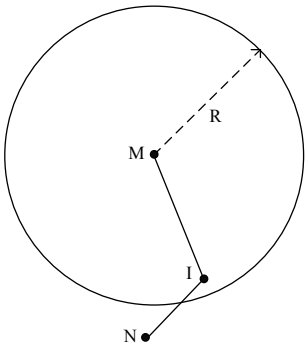


Figure 1.7 – Principle of distance estimation in Sum-Dist

To estimate its position, an unknown node u draws, for each anchor a, one or two circles:

- If u and a are neighbours, then u deduces that it is on the circle centred in a and of radius d_{au}.
- If u and a are not neighbours then u deduces that it is not inside the circle centred in a and of radius R. In addition, u knows the estimated distance $\widehat{d}_{au}$. Since $d_{au} \leq \widehat{d}_{au}$

(triangular inequality) then u deduces that it is inside the circle centred in a and of radius $\widehat{d}_{au}$.

The intersection of these circles provides a zone Z_u containing u. The unknown node u estimates its position as the centroid of this zone.
If we note:

- $\mathcal{A}$ the set of anchors.
- $\mathcal{N}_{\mathcal{A}}(u)$ the set of neighbour anchors for an unknown node u.
- $\bar{\mathcal{N}}_{\mathcal{A}}(u)$ the set of non-neighbour anchors for an unknown node u.
- $\mathcal{X}$ the set of all possible coordinates in the network.

For each unknown node u, Z_u is obtained as follow:

$$Z_{\mathcal{N}_{\mathcal{A}}(u)} = \bigcap_{a \in \mathcal{N}_{\mathcal{A}}(u)} \left\{ (x_i, y_i) \in \mathcal{X} \middle| (x_i - x_a)^2 + (y_i - y_a)^2 = d_{ua}^2 \right\}$$

$$Z_{\bar{\mathcal{N}}_{\mathcal{A}}(u)} = \bigcap_{a \in \bar{\mathcal{N}}_{\mathcal{A}}(u)} \left\{ (x_i, y_i) \in \mathcal{X} \middle| R^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \leq \widehat{d}_{ua}^2 \right\}$$

$$Z_u = Z_{\mathcal{N}_{\mathcal{A}}(u)} \bigcap Z_{\bar{\mathcal{N}}_{\mathcal{A}}(u)}$$

An example is displayed in Figure 1.8. The unknown node X first uses SumDist to estimate the distances $\widehat{d}_{XA}$, $\widehat{d}_{XB}$ and $\widehat{d}_{XC}$ to the different anchors A, B and C. A, B and C are not neighbours of X, X is consequently not inside circles centred respectively at A, B, C and of radius R but it is inside circles of radius $\widehat{d}_{XA}$, $\widehat{d}_{XB}$ and $\widehat{d}_{XC}$. The correlation of these information defines the red hatched area Z_X. X estimates its location as the center of gravity of Z_X. Like most of the traditional multi-hop localization schemes (e.g., Dv-hop), AT-Dist suffers from high communication overheads in dense networks.

Mobile element assisted algorithm: [Chen 10] proposes a mobile anchor assisted localization technique for static wireless sensor networks. The mobile anchor is assumed to be more powerful than the deployed unknown nodes and have M levels of transmission power with the corresponding transmission radii R_i, $i = 1, 2, ..., M$. At different points of its trajectory, the mobile anchor consecutively broadcasts beacons at varying power levels including its current position, considered transmission power and transmission radius. Unknown static nodes use the received beacons to construct constraints on their locations. Indeed, if an unknown node at position x receives a beacon from the mobile anchor which current position is a and considered transmission range is R then it will conclude that its distance to the anchor verifies $\|x - a\| \leq R$. Otherwise the distance verifies $\|x - a\| > R$.

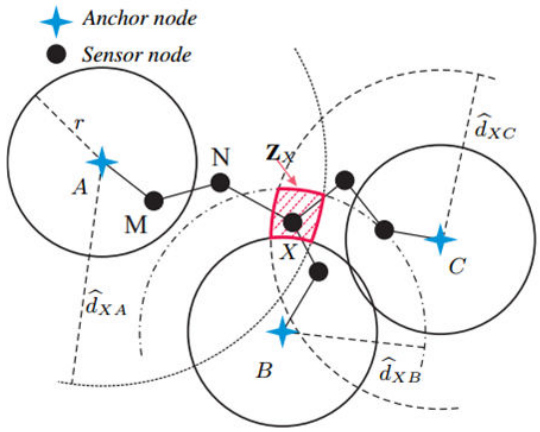


Figure 1.8 – Principle of the localization process in AT-Dist

Based on the multiple transmission powers of the mobile beacon at each position, the unknown node derives the following inequalities: $r_k < \|x - a_k\| \leq R_k,\ k = 1, 2, ..., n$ Where a_k is the mobile anchor position at time k, R_k and r_k are valid transmission radii at time k. The problem of determining the unknown nodes positions is hence transformed into the resolution of a set of quadratic inequalities.

PI: paper [Guo 10], proposes a mobile anchor assisted localization algorithm called Perpendicular Intersection (PI). PI is tradeoff between range-free and range-based approaches. Instead of directly mapping RSSI absolute values to distances, PI first utilizes RSSI values to only approximate the closeness degree between the mobile anchor and the unknown nodes and then uses the geometric relationship of perpendicular intersections to compute nodes locations. The proposed algorithm was motivated by the observation that the closer a node to the signal sender, the higher the RSSI value that it receives. That is, although the irregularities of the RSSI, it is generally a fact that the RSSI values between two nodes monotonically increase as the nodes get closer to each other. Experiments with a mobile anchor A moving on a straight line and broadcasting signals have shown that the highest RSSI value perceived by a neighbouring sensor node usually corresponds to the point on the line that is closest to the node. Theoretically, this point corresponds to the projection of the node on the anchor straight line trajectory. Using two projections of the sensor node on two straight line trajectories, the node position is estimated as the intersection point of the two perpendiculars that cross the mobile anchor trajectories at the two projections, respectively. Figure 1.9 illustrates how PI works. The mobile anchor starts its trajectory at point A_1 where it broadcasts a start beacon containing its current position. The unknown node X is within the communication range of the mobile anchor, it then receives and registers the start position $A_1(x_1, y_1)$. The anchor linearly moves from

A_1 to A_2 and periodically broadcasts beacons with its current position. Node X receives the beacons and registers the beacon with the highest RSSI value. The mobile anchor broadcasts a stop beacon when it reaches point A_2. Node X receives the stop beacon and notes that the mobile anchor has finished its trajectory from A_1 to A_2. At the end of the trajectory A_1A_2, X will have recorded the position $X'(x', y')$ where the beacon packet has the highest RSSI value. According to the authors observations, this position corresponds to the projection of X on the line A_1A_2. Similarly, at the end of the anchor trajectory A_2A_3, node X will have recorded a new position X'' which correspond to its projection on line A_2A_3. Having the coordinates of $X'(x', y')$, $X''(x'', y'')$ and $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$, the unknown node X can compute its coordinates as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{pmatrix}^{-1} \times M$$

Where

$$M = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 & 0 & 0 \\ 0 & 0 & x_3 - x_2 & y_3 - y_2 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ x'' \\ y'' \end{pmatrix}$$

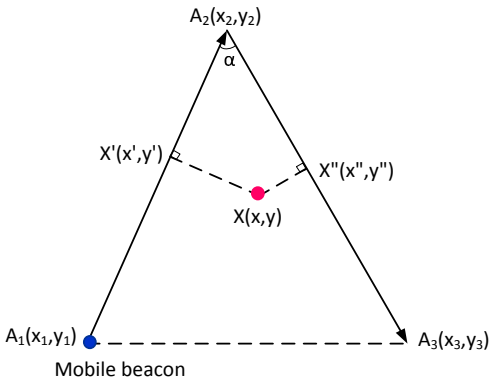


Figure 1.9 – PI principle

To guarantee that all the unknown nodes within the virtual triangle $\Delta A_1A_2A_3$ can receive the mobile anchor beacons, the lengths A_1A_2 and A_2A_3 should be shorter than the anchor communication range and the angle α between the two lines A_1A_2 and A_2A_3 should satisfy $0 < \alpha \leq \frac{\pi}{3}$. As depicted in Figure 1.10, the optimal trajectory of the mobile anchor consists of multiple equilateral triangles with side length R covering the hole deployment area.

APIT: in [He 03], He et al. present the Approximate Point In Triangle (APIT) technique. APIT requires a heterogeneous network where anchor nodes are equipped with

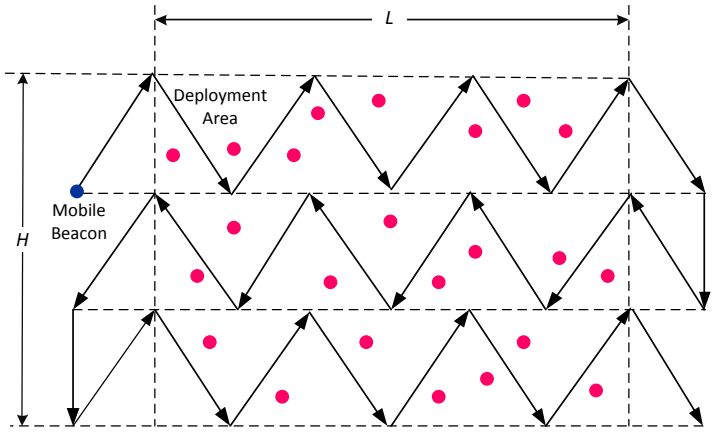


Figure 1.10 – Optimal trajectory of the mobile beacon

high-powered transmitters. An unknown node first forms a number of *reference triangles*. A *reference triangle* is the triangle formed by connecting three reference nodes. It then, using the Point In Triangle (PIT) algorithm [He 03], tests whether it is inside or outside a given triangle. The PIT algorithm is based on the following two propositions:

Proposition 1: if M is located in the triangle ABC, then, when M moves towards any direction, its new location must be nearer to (further from) at least one anchor A, B or C.

Proposition 2: if M is located out of the triangle ABC, then, when M moves towards any direction, there must exist a direction in which the location of M is further from or closer to all triangle's three point A, B and C.

Figure 1.11 shows the principle of the PIT algorithm.

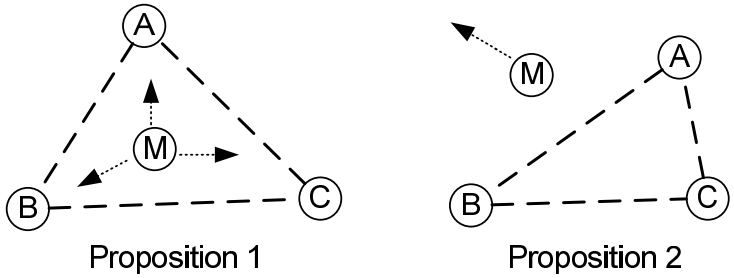


Figure 1.11 – The principle of the PIT algorithm

The estimated position of the unknown node is the center of the reference triangles that contain it. APIT is easy to implement and has low computational complexity. Nevertheless, its performance deeply depends on the density of the anchor nodes. A few number of anchors does not allow enough triangular regions to overlap, and in this case the accuracy of the algorithm will decrease.

Polygon-Based algorithm: paper [Datt 06] proposes a range-free localization algorithm for both static and mobile sensor networks. It is assumed that sensor nodes are aware of their maximum velocity v_{max}. Each node maintains a polygon representing the set of its probable current locations. At every time-step, each node (anchor and non-anchor nodes) broadcasts a beacon packet with its polygon dilated by the communication range R. When an unknown node receives the polygon of a neighbouring node, it dilates its own polygon by d_{max} (the maximum distance it can move in a time-step) and computes the intersection of the two polygons. The resulting intersection polygon is the new location polygon for the node.

1.4 Conclusion

In this chapter, we presented an overview and a taxonomy of the localization techniques in wireless networks. We classified localization techniques into different categories depending on their finality, architecture, used signal technology, ranging assumption, infrastructure assumption, application environment and nodes mobility. We then cited the major representative works in each category. For a deeper understanding of the localization problem, we also detailed a selected set of typical localization algorithms.

Chapter 2

Conflict Detection and Resolution in Mobile Wireless Networks: State of the Art and Foundations

Contents

This chapter presents the state of the art and the foundation of conflict avoidance for mobile wireless networks. A conflict (not to be confused with a collision) is a situation

in which two or more mobile nodes experience a loss of minimum separation [Kuch 00]. Such conflicting situations should be avoided in order to achieve safety and prevent the collision risks between the nodes. The conflict avoidance process is generally organized into three fundamental functions: the sensing function, the detection function and the resolution function.

- Sensing: the sensing function represents the capability of the mobile node to perceive its surrounding environment and collect current state information for encounters. Using the appropriate communication equipment and sensors, a mobile node can have an estimation of the current traffic situation (e.g., nearby nodes positions and velocities).

- Detection: the detecting function enables the mobile nodes to discover future conflict risks. First, state information is projected into the near future. Current and future state information are then combined to extract conflict metrics (e.g., closest point of approach). Using conflict metrics, a decision is finally made as to whether an actual risk of conflict exists and if an avoidance manoeuvre is needed.

- Resolution: when a near future conflict is detected, the conflict resolution function may be invoked. The main role of the resolution function is to avoid a possible collision with an encounter by determining how and which manoeuvres should be performed. Once the collision risk has been mitigated by the appropriate avoidance actions, the mobile node can return back to its original course path. Conflicts should be detected in a sufficient time beforehand so as to provide enough time for the determination and the performance of the avoidance manoeuvres.

Figure 2.1 shows the different steps of the conflict avoidance process. These steps can be implemented in different ways, giving rise to several technical solutions. These solutions can be classified based on different design factors. Figure 2.2 summarizes the major design factors by which approaches differ. These factors will be discussed in details in the following sections.

Next, we will use the term *mobile node* or *vehicle* to refer to any object capable of navigation within its environment (e.g., an airplane). During its cruise, a mobile node may be fully autonomous, semi autonomous or completely guided by an operator. Depending on its autonomy level, the role of a mobile node may range from the simple sensing to the fully autonomous sensing, detection and resolution.

The rest of this chapter is organized as follows. Section 2.1 details the sensing function and reviews the currently used sensing technologies. Section 2.2 details the detection function and classifies the conflict detection approaches according to different aspects.

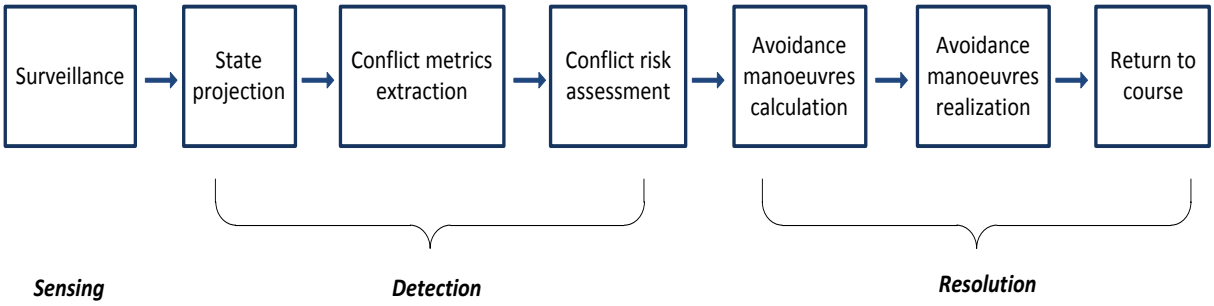


Figure 2.1 – Conflict avoidance process

Section 2.3 details the resolution function, categorizes the corresponding researches based on four design aspects and argues the advantages and disadvantage of each category. Section 2.4 concludes the chapter.

2.1 Sensing Technologies

Surveillance for conflict avoidance can be achieved by using a wide variety of sensors that can be divided into two main categories: cooperative and non-cooperative sensors. Cooperative sensors comprise all communication equipment that enable the nodes exchange their mobility data. Nodes not fitted with such communication equipment may instead rely on non-cooperative sensors. In this case other surrounding nodes are directly sensed, irrespective of their desire to be sensed.

2.1.1 Cooperative sensors

Cooperative conflict avoidance techniques rely on a set of devices permitting information exchange, such as position, heading, speed and waypoints, between the mobile nodes. Transponders and the ADS-B technology are examples of cooperative sensing techniques.

Transponders (contraction of the words transmitter and responder) are electronic devices that transmit a specific reply when receiving a specific radio frequency interrogation. This sensing method performs well in controlled spaces where all the mobile nodes are fitted with transponders. However it does not permit the sensing of non-transponding targets, so such targets have to be sensed via other means.

The Automatic Dependent Surveillance-Broadcast (ADS-B) is a technology used by airplanes allowing them to detect other similarly equipped aircraft with much more precision than transponders. In ADS-B, an aircraft determines its precise 3D position using the Global Navigation Satellite System (GNSS). The aircraft position, along with other information such as its unique identifier, speed and heading intent are periodically updated

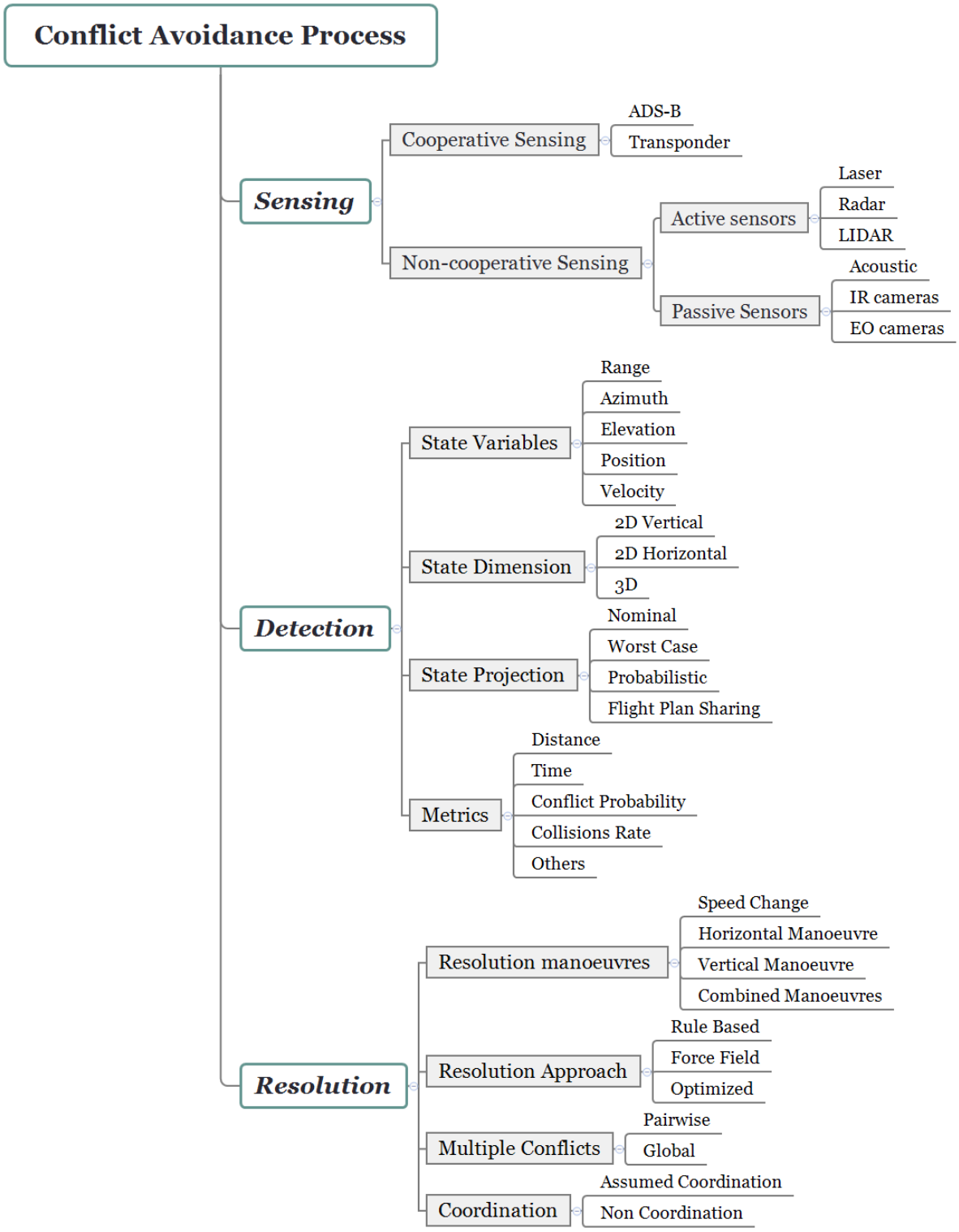


Figure 2.2 – Design factors of conflict avoidance techniques

and broadcasted via data links. The resulting periodic and widely available position feedback enables accurate monitoring of the aircraft by the other airplanes in its surrounding

airspace. ADS-B heavily depends on the integrity of the aircraft navigation system. The failure of this system prevents the airplane from broadcasting its accurate position. Another drawback of the ADS-B is that it is relatively easy to simulate non-existent traffic by sending fraudulent messages.

2.1.2 Non cooperative sensors

Non cooperative sensors are promising technologies since they, unlike cooperative sensors, do not require the coordination with similarly equipped nodes. Moreover they benefit from the fact that they can be used to detect both moving and stationary obstacles. Non cooperative sensors can be divided into two categories: active or passive sensors. Active sensors diffuse signals to discover obstacles in the movement path while passive sensors rely on the signals emitted by the obstacles themselves.

Examples of active sensors are the laser range finder, the radar and the LIDAR. Laser range finder is a device that uses laser energy to estimate the distance to an object. A laser pulse is first sent in a narrow beam towards the target. Then, the propagation time of the pulse to the target and back is measured to determine the source-target distance. Most laser range finders are planar. That is obstacle below or above the measurement plane are not detectable. Alternatively, radar systems can be used to detect any enclosing object. A radar uses radio waves to estimate the position and the speed of an object. It is able to detect mobile objects, terrains and even weather formations (rain, snow, hail, etc). A radar has a transmitter that emits radio waves that are reflected by any object in their path and detected by a receiver. This technique is ideal in bad optical vision conditions such as stormy, foggy and night times. However, it is not used in small vehicles because of its large weight and size. Moreover radar systems are considered costly. LIDAR, which stands for Light Detection and Ranging or Light radar uses light in the form of a pulsed laser in order to measure distance to a target. These light pulses combined with other data, such as the emitter position, generate precise, 3D information about the target.

Passive sensors mainly include acoustic sensors, Electro-Optical (EO) cameras and Infra-Red (IR) cameras. Acoustic sensors are used for detecting and tracking objects only by their emitted sounds. They generally involve a simple and light hardware and cover only a short range. EO cameras provide mobile nodes with day-light visions by recording the reflected light. IR cameras produce night visions by detecting the objects heats. Cameras are small, light and inexpensive and therefore suit small vehicles. They are able to provide a wide field of view with high resolutions, but on the other hand, this leads to a significant data processing. They can also give precise information about the azimuth and the elevation angles. Range information is not directly provided, it must be

either inferred or sensed using other means.

Table 2.1 summarizes the above mentioned sensing technologies. Essentially, a sensor needs to acquire the range, the azimuth and the elevation (Figure 2.3) of the objects of interest [Saha 15]. Most of the cooperative sensing techniques are able to accurately provide these information, but they are only capable of surveying alike equipped nodes. Non cooperative sensing techniques can detect both cooperative and non cooperatives objects. However, they usually operate over a much shorter distance than cooperative sensors. Distant objects can't be correctly sensed. Besides, some non cooperative sensors, such as lasers and cameras, are only capable of providing either range or azimuth and elevation information but not both. The ability to detect obstacles at night and in bad weather conditions are also important attributes evaluating the sensing technology. All the mentioned sensing techniques, except the EO cameras, are capable of detecting obstacles day and night. However, only transponders, ADS-B and radars are effective in bad weather conditions.

Due to the restrictions in each sensing mode, a single sensor cannot assure a complete monitoring. Multiple sensors are required to provide an exhaustive solution. Thus, the weakness of a sensor can be compensated by the strength of another.

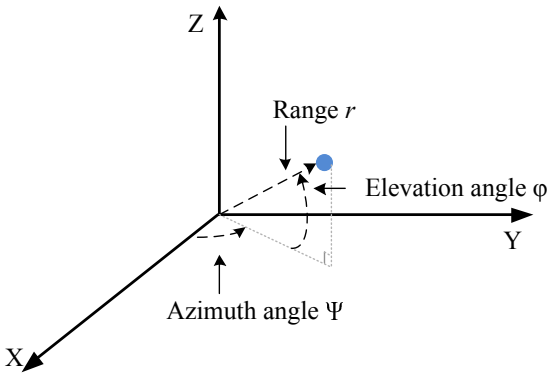


Figure 2.3 – Main sensing information

2.2 Conflict Detection

A conflict occurs when the distance between two or more nodes is less than a minimum required separation distance. This separation requirement defines a protected zone surrounding each node that should not be infiltrated by any other node. For example, in a 3D plan, a minimum separation distance of D_{min} results in a spherical protected zone of radius D_{min} around each mobile node. The minimum separation could also be divided into a horizontal minimum distance and a vertical minimum distance resulting in a cylindrical protected zone around each node. The protected zone may also be defined in

Technology	Category	Range	Azimuth	Elevation	Day & Night	Bad Weather
Transponder	Cooperative	Yes	Yes	Yes	Yes	Yes
ADS-B	Cooperative	Yes	Yes	Yes	Yes	Yes
Laser	Non-Cooperative (Active)	Yes	No	No	Yes	No
Radar	Non-Cooperative (Active)	Yes	Yes	Yes	Yes	Yes
Acoustic	Non-Cooperative (Passive)	Yes	Yes	Yes	Yes	No
EO camera	Non-Cooperative (Passive)	No	Yes	Yes	No	No
IR camera	Non-Cooperative (Passive)	No	Yes	Yes	Yes	No

Table 2.1 – Sensing technologies

terms of parameters other than the minimum separation distance such as the minimum separation time. We define conflict detection as the process of detecting future conflicting traffic. Conflict alerts must be issued early enough that avoidance manoeuvres can be performed, but not very early that nuisance warnings occur. The conflict detection approaches mainly differ by:

- The considered state information
- The state projection method
- The alerting metrics

In followings, we first discuss each of these three factors and then define the primary metrics that quantify the reliability of a conflict detection approach.

2.2.1 State information

State information provides an estimate of the current traffic situation. It may be composed by one or many state variables such as the range r, the azimuth angle ψ, the elevation angle ϕ, the speed v, the acceleration a, the position vector $\mathbf{x}$ and the velocity vector $\mathbf{v}$. Depending on the used sensing technology, this information can be acquired directly or indirectly and may cover the two dimensional vertical plane ($2D_V$), the two dimensional horizontal plane ($2D_H$), or the three dimensional plane ($3D$). The $3D$ and the $2D_H$ position and velocity vectors are the most used state information to describe the surrounding environment [Dowe 07, Muno 13].

2.2.2 State projection

State projection shows how the current states are projected into the future. There are mainly three projection methods : nominal, worst case and probabilistic [Kuch 00].

In the nominal projection method [Alba 09, Yang 13], the current states are propagated into the future along a single trajectory (e.g., straight trajectory). It completely ignores uncertainties in the state projections. The nominal projection is simple but it is only applicable when the mobile nodes trajectories are very predictable or when the propagation period is very short. Indeed, this approach does not take into account the fact that an encounter may not behave as predicted, an aspect that is very probable in a long propagation period.

In the worst case projection [Toml 98], a whole range of possible future manoeuvres is examined. If any one of these manoeuvres could cause a conflict, then a conflict alert is triggered. This approach detects conflict risks even in worst case scenarios however, it is far from providing optimal solutions. It should be limited to a short propagation period to reduce the computation requirement and avoid excessive false alerting.

The probabilistic projection [Liu 10, Chry 11, Saha 14] considers uncertainties in the mobile nodes future trajectories. This uncertainty can be modeled either by adding errors to the nominal trajectories and then deriving conflict probability or by building a full spectrum of future trajectories and assigning a probability of occurring to each one (e.g., using a probability density function). A conflict alert is issued if the conflict probability is above a given threshold. The nominal and the worst case projection methods are subsets of the probabilistic approach: in the nominal case the mobile node follows a single trajectory with probability one; in the worst case the mobile node may follow any trajectory with equal probability. The probabilistic projection causes less missed alarms than the nominal projection and less false alarms than the worst case projection. Nevertheless, methods using this approach usually require heavy processing. It may also be difficult to model the probabilities of future trajectories.

The projection may be based only on current state information or may also use additional information such as a trajectory plan [Sisl 06, Yepe 07, Hwan 08]. The trajectory plan describes the mobile node future waypoints along its path. This intent information can be used to ameliorate future trajectory estimation and consequently enhances the alerting decisions.

2.2.3 Conflict metrics

Conflict metrics are the parameters derived from current and predicted states to make alerting decisions. Unlike the state projection that can be separately performed for each node, conflict metrics extraction necessitates the aggregation of the states of the different involved nodes. The minimum separation distance, the time to the closest point of approach [Muno 10], the number of available avoidance manoeuvres [Yang 97], the con-

flict probability [Paie 97] and the conflict rate [Leve 11] are the most used conflict metrics. Some approaches use more complex and direct metrics than physical metrics. In [Yang 02] the authors propose an alternate approach where alerting decisions are based directly on computed values of performance metrics such as false alarms probability.

Based on the derived metrics, a decision is made on whether an avoidance manoeuvre is needed to avoid the collision menace. This decision is usually made by comparing conflict metrics with one or more threshold values. The threshold values depend on many parameters such as the surrounding environment, the vehicles states, characteristics and performances. Ideally, the threshold values should be dynamically adapted to the specific monitored situation. For example paper [Isaa 97], dealing with conflict detection and resolution for airplanes, considers different alert thresholds depending on the flight altitude and whether the airplanes are in a descent or a climb trajectory.

2.2.4 Measures of reliability

The reliability of a conflict detection approach can be mainly measured in terms of false alarms and missed alarms [Alam 09]. A false alarm is an issued alert without a subsequent conflict. Conversely, a missed alarm is a conflict with no prior issued alert. Missed alarms are considered as an extreme hazard leading to perilous collisions while false alarms are considered as nuisance alarms resulting in unnecessary escape manoeuvres. Hence, minimizing false alarms and eliminating missed alarms is a crucial design requirement for conflict detection algorithms.

2.3 Conflict Resolution

The conflict resolution function should be invoked once a near future conflict is detected. We define the conflict resolution as the process that specifies how a particular conflicting situation can be resolved in order to avoid an imminent collision. Conflict resolution techniques can be mainly categorized based on four design aspects which are:

- The used resolution manoeuvre
- The considered resolution approach
- The handling of multiple nodes conflicts
- The assumption of coordination or non coordination between the nodes

In this section, we explain each of these aspects as well as the key measures of effectiveness of a conflict resolution technique.

2.3.1 Resolution manoeuvres

The resolution manoeuvres are the set of actions used to resolve a conflict. Basic manoeuvres include speed change (speedup or slowdown), horizontal manoeuvre (turn left or right) and vertical manoeuvres (climb or descend). In some cases a single basic manoeuvre is sufficient to resolve a conflict. In others a combination of basic manoeuvres is required . The combined manoeuvres may be performed simultaneously or sequentially.

2.3.2 Resolution approach

Resolution approaches are the methods by which resolution manoeuvres are generated. There are three major resolution approaches categories: rule based, force field and optimized.

In the rule based approaches [Carp 97, Luo 13, Kuwa 14] conflicts are resolved based on a set of pre-defined rules. In[Carp 97], when a conflict alert is issued, the threatened airplane is assumed to perform a fixed climbing turn avoidance manoeuvre to turn away from a parallel moving intruder. Rule based approaches are generally easy to implement and simple to understand which reduces the response time to conflict alerts. However, they usually do not appropriately account for unexpected events in the environment. For example, in [Carp 97] the climbing manoeuvre provides a vertical separation between the threatened vehicle and the intruder. If the intruder unexpectedly climbs, this vertical separation would be reduced or even eliminated resulting in an additional conflict.

Force field approaches [Wen 12, Saha 13, Carl 13, Reza 14, Choe 14] consider nodes as charged particles evolving in a force field and use the repulsive forces between them to generate escape manoeuvres. The force field approaches use relatively simple electrostatic equations to resolve conflicts, but the feasibility of the computed solutions is not guaranteed due to the vehicles dynamic limitations. A force field solution may for example require a sharp variation of the vehicles direction or speed which is physically very difficult or infeasible.

Optimized approaches produce resolution manoeuvres that minimize a certain cost function such as trajectory duration, energy consumption, deviation from the original trajectory or workload. Optimization approaches include different sub-categories that differ in the method by which the resolution decision is derived. Main sub-categories include game theory based methods, genetic based methods and optimal control theory methods. In the game theory based methods, the conflict resolution problem is formulated as a cooperative [Arch 08] or a non-cooperative [Toml 98] game. Genetic based methods [Mond 01] generate optimized resolution manoeuvres using techniques inspired by natural evolution such as crossing, mutation and selection. In optimal control theory methods

[Hoff 08] the nodes dynamics, the constraints and the cost functions are defined and an optimal resolution is determined. Optimized approaches seem attractive since they minimize costs. Nevertheless, the complexity of the used functions can make the optimized approaches difficult to understand and computationally intensive.

2.3.3 Management of multiple nodes conflicts

There are two approaches by which a conflict resolution technique can handle conflicts between more than two nodes: Pairwise and Global. In the pairwise approach conflicts are addressed sequentially in pairs. In the global approach the whole situation is assessed simultaneously and the conflict is resolved at once. This is usually done by grouping all the nodes involved in the conflict in a cluster. Although the pairwise approach is much simpler than the global approach, it may, in some situations, lead to unsafe or ambiguous situation. Figure 2.4(a) shows a conflict between a set of airplanes in which a pairwise solution is unable to solve the conflict problem. Aircraft A detects a future conflict with aircraft B and decides to climb or to dive, at a fixed threshold time T before contact, in order to avoid the collision with B. This solutions is unsafe since it generates a new conflict either with airplane C or with airplane D. Airplane A will not have enough time to resolve the new generated conflict and can't consequently avoid the collision. In Figure 2.4(b), a global solution considers the three menacing airplanes simultaneously and realizes that the climbing or the diving manoeuvre must be initiated earlier in order to safely resolve the conflict. Global approaches may offer more robust solutions than the pairwise approaches, however they require a lot of computational complexity.

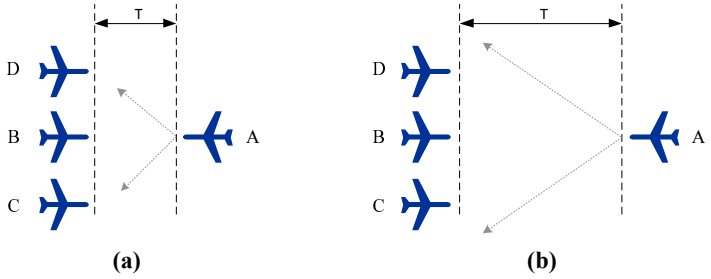


Figure 2.4 – Pairwise(a) vs. Global(b) conflict resolution

2.3.4 Coordination

Assumed coordination or non-coordination between conflicting nodes is one of the important design factors affecting the conflict resolution process. In coordinated conflict

resolution, involved nodes cooperatively agree on a set of manoeuvres in order to resolve conflicts. In non-coordinated conflict resolution, each node exclusively manoeuvres without taking into consideration the other involved nodes eventual manoeuvres.

Several coordination strategies have been proposed. Some strategies applied multi-agent systems theory, others used rule based, code number or token allocation methods. Methods based on multi-agent systems theory [Sisl 07] model mobile nodes as intelligent agents and establish a communication protocol that allows the coordination and the negotiation of resolution actions between the different agents. Rule based coordination [Duon 96, Dowe 05] defines implicit priorities to nodes involved in the conflict. Unlike the agent based methods, explicit decision coordination between the nodes is not necessarily required. Instead, rule based coordination may require different kinds of information about the conflicting nodes such as the position vector, the velocity vector and the intent information. In [Duon 96], the current position and velocity vectors of the vehicles is used to coordinate between conflicting traffic: if the tracks of two vehicles cross, the vehicle with the other to its right have to give way and pass behind. The code number and the token based coordination methods [Gran 01] are used to enforce a global resolution priority order between the conflicting nodes. Coordination helps reduce manoeuvring cost and avoid manoeuvres that would intensify or extend the conflict. However, coordination may be interrupted in case of data links failure or data exchange interference. Thus, a conflict resolution method with assumed coordination should also be able to handle cases in which coordination is impossible.

2.3.5 Measures of effectiveness

The number of near misses and the resolution cost are generally used as the primary metrics to quantify the performance of conflict resolution techniques. From safety point of view, any conflict resolution technique is required to maintain a minimum separation distance between the nodes. Any violation of this safe separation results in a near miss. Conflict resolution techniques should ensure as few near misses as possible. The resolution cost evaluates the loss produced by the resolution manoeuvres. Usually, a set of basic cost parameters are combined to generate a cost function estimating the loss yielded for the nodes. The basic cost parameters essentially include the additional energy consumption, the delays at planned waypoints, the number of necessary resolution manoeuvres and the total heading/altitude changes.

2.4 Conclusion

In this chapter we discussed in details the avoidance process for mobile nodes. Conflict avoidance is usually organized into three basic functions: the sensing, the detection and the resolution function. We explored each of these functions and reviewed the most noteworthy technologies and approaches treating each function.

Chapter 3

Weighted Localization in Mobile Wireless Networks

Contents

In this chapter, we propose ASAW a new ASynchronous, Aggregate Weighted localization algorithm for mobile wireless networks. Taking into consideration the drawbacks of the previously proposed localization techniques, we propose next a localization algorithm that:

- Does not require the synchronization of nodes. In order to avoid the restrictive synchronization requirement and collisions due to synchronized transmissions, we assume that nodes broadcast their locations at different non-synchronized times and normal nodes continuously record the received localization information. A normal node can, at any time (and not at a precise time step), use the recorded location information in order to estimate its position.

- Aggregates different metrics of different types (e.g, link quality, distance, velocity, etc) in order to weight the received location information providing as such a better reliability and robustness against the uncertainty of certain metrics.
- Is independent from the nodes radio transmission range.
- Requires a low computational cost (a small number of basic mathematical operations) and a low communication overhead (only 1-hop messages broadcasting is required) which enables its implementation in resource-constrained nodes.
- Uses both neighbouring anchor and normal nodes location information and hence does not require an increased anchor density. Normal nodes location estimates are attributed a parameter ϵ quantifying their quality. Such a parameter is used in order to safely use the normal nodes location estimates.

The chapter is organized as follows. Section 3.1 reviews the previously proposed weighted localization algorithms and highlights their major problems. Section 3.2 presents ASAW our novel weighted localization algorithm. Section 3.3 uses simulations to evaluate the performance of ASAW and compare it with two widely used localization techniques. Section 3.4 concludes the chapter.

3.1 Weighted Localization Techniques

Paper [Bulu 00] proposes the Centroid Localization (CL) algorithm. In CL, the estimated location $\hat{x}$ of an unknown node u is calculated as the centroid of the coordinates X_i of beacon nodes within its communication range, that is:

$$\hat{x} = \frac{\sum_{i=1}^{k} X_i}{k}$$

Where k is the number of all in-range beacons. Instead of using the coordinates of all in-range beacons, in [Zou 14] an unknown node first collects the RSSI of all the nearby beacons, selects those whose RSSI is above a given threshold and finally estimates its location as the average of these chosen beacons. One big issue with such centroid localization techniques [Bulu 00, Zhan 13, Qian 14, Zou 14, Meng 14] is that they assume that all the selected reference points are equally proximate to the unknown node [Piva 11]. Since such an assumption is usually not satisfied in practice, the authors of [Blum 07] introduced the Weighted Centroid Localization (WCL) algorithm where each reference point is attributed a weight depending on its distance to the unknown node. An unknown node position is

estimated as the weighted average of the coordinates of known reference points:

$$\hat{p} = \frac{\sum_{i=1}^{k} w_i \; X_i}{\sum_{i=1}^{k} w_i}$$

The parameter w_i is the weight of the i^{th} reference point and is equal to $\frac{1}{(d_i)^g}$. The distance d_i is the distance between the unknown node and the i^{th} reference point and is estimated through the RSSI received from the reference point. The parameter $g > 0$ determines the weight of the contribution of each reference point. If $g = 0$, then the WCL is reduced to the CL approach. Increasing the value of g increases the weight of the closest reference points. In case of very high values of g, the estimated position moves to the closest reference point position and the positioning error increases [Blum 07]. The authors used simulations in order to determine the optimal value of g. They found that the optimal value of g depends on the nodes transmission ranges and the dimension of the network and a degree $g = 3$ yields in best results in a network of $30m$ x $30m$ and a transmission range $R = 30m$.

Many recent works have adopted the WCL approach [Yang 10, Wang 11b, Hai 11, Chau 16]. The paper in [Yang 10] proposes RR-WCL, a weighted centroid localization algorithm using RSSI ratio. In the RR-WCL algorithm, anchor nodes periodically send their location information and the static unknown nodes only record the RSSI mean values received from each anchor. When the number of received beacon messages reaches a given threshold then an unknown node uses the RSSI ratios as weights in order to determine its position. Paper [Hai 11] proposes IWCA, an improved weighted centroid localization algorithm based on the messages travel times as weights. Beacon nodes broadcast their location information and unknown nodes receive and estimate the messages travel times. Beacon nodes with shorter travel times are considered closer to the normal node and are hence attributed higher weights. The proposed approach is simple. However, calculating the messages transmission times requires that all the involved nodes (anchors and normal nodes) are synchronized and that the messages sending times are labeled in each sent packet. Most of the proposed WCL approaches exclusively rely on a single parameter, and especially the RSSI, to weight the collected location information. Depending on a single metric can nevertheless result in poor position estimations particularly when the considered metric is not sufficiently reliable (RSS is unstable in real environments). Combining several measures from different categories would provide better performances than just relying on a single metric.

The works in [Hu 04, Ruda 07, Zhan 10b, Sheu 10, Huan 15] propose Monte Carlo

Localization (MCL) algorithms. In MCL techniques each unknown node maintains a set of weighted samples representing its possible positions and estimates its position as the weighted average of these samples. Time is divided into discrete time steps. Location samples are updated at each time step based on the samples in the previous time step and are attributed weights depending on the current observations and current state of the system. In [Hu 04], each node uses the positions of its neighboring anchors to weight its samples. The weight of each sample is either 0 or 1. Relying only on the anchors location information requires an increased anchor density in order to achieve reliable location estimates. Anchor nodes are yet generally more expensive and are deployed in much lower densities than normal nodes. It will therefore be very advantageous if the estimated locations of the normal nodes can also be used to improve the localization accuracy. The work in [Ruda 07] have hence extended [Hu 04] by using the location estimates of non-anchor (normal) neighbors and not just anchor nodes. Nodes use only the information of normal neighbors that have more accurate estimates than theirs. The quality of a position estimate is measured using a parameter called *closeness*. The closeness value for a node q with N samples is computed as follows:

$$closeness_q = \frac{\sum_{j=1}^{N} w_j \sqrt{(x_j - x)^2 + (y_j - y)^2}}{N}$$

Where (x_j, y_j) is the coordinate of the *j-th* sample, w_j is the weight of the *j-th* sample and (x, y) is the current position estimate of node q. The closeness for an anchor node is zero (anchor nodes locations are assumed to be perfect) while the closeness parameter for a non anchor node is greater than zero. Higher closeness values indicate lower accuracy in the position estimate. Paper [Zhan 10b] uses a bounding-box that improves the sampling efficiency by reducing the scope from which the samples are selected. Besides a normal node can estimate its maximum localization error in the x-axis (ER_x) and in the y-axis (ER_y) using its position estimation and its bounding-box. If we note (x_e,y_e) the normal node estimated position and (x_{min}, x_{max}, y_{min}, y_{max}) its bounding-box then ER_x is equal to $max(x_e - x_{min}, x_{max} - x_e)$ and ER_y is equal to $max(y_e - y_{min}, y_{max} - y_e)$. To estimate the unknown nodes locations, the proposed algorithm uses 1-hop neighboring anchors and normal nodes as well as the 2-hop neighboring anchors location information. Considering 2-hop beacon broadcasting may ameliorate the location estimations but will on the other hand increase the communication costs particularly in high density networks.

The major disadvantages of MCL techniques is that they require the knowledge of the nodes radio ranges and assume that nodes are synchronized and can send and calculate their location information at the same discrete time step. In real environments, the radio ranges are nevertheless constantly changing due to different factors including the nodes

residual energy and surrounding environment. Besides, time synchronization is generally a difficult task to achieve in wireless networks. Finally, the synchronized sending of location information increases the probability of packets collisions and hence the loss of the location information. All these issues with the previously proposed localization algorithms have inspired us to propose ASAW. Indeed ASAW, is asynchronous, independent from the nodes radio range and uses the aggregate weight of different metrics instead of relying on a single metric.

3.2 ASAW: an ASynchronous, Aggregate Weighted localization algorithm

This section presents our new localization algorithm. We first detail how a normal (unknown) node calculates its aggregate weighted location. Particularly, a normal node uses a set of basic mathematical operations with a low computational cost such as additions, substractions and multiplications. We then describe the behaviour of both anchor and normal nodes within the network. Nodes rely only on 1-hop communication (no flooding) which minimizes the communication overhead.

3.2.1 Location estimation

The system consists of three categories of nodes: fixed anchor nodes, mobile anchor nodes and mobile normal nodes. Both anchor and normal nodes broadcast messages with their location information. A location information message is as follows: *Loc_msg(ID, (x,y,z)*, v, a, ϵ). ID is the identity of the sender; (x,y,z) is the location estimate of the sender; v is the velocity of the sender; a is set to 1 if the message sender is an anchor and to 0 if the message sender is a normal node. The parameter ϵ describes the quality of the sender location estimate. The higher is ϵ the better is the quality of the location estimation. If the message sender is an anchor node then $\epsilon = 1$. If the sender is a normal node then $\epsilon \leq 1$.

Each normal node maintains a location information table (*Loc_tab*) in which it stores the received location information (Figure 3.1). A normal node updates its location information table for each received location information message. As shown in algorithm 1, once a location information message is received, the normal node first estimates the distance to the sender d, the distance estimation error Δ, the link quality Q and then records the received location information message, the estimated d, Δ, Q as well the time of the reception of the message t_r in the location information table.

At any time, a normal node can estimate its location as the weighted average of the

Sender ID	Position (x, y, z)	v	a	ϵ	d	Δ	Q	t_r
1	(x_1, y_1, z_1)	v_1	1	ϵ_1	d_1	Δ_1	Q_1	t_{r_1}
2	(x_2, y_2, z_2)	v_2	0	ϵ_2	d_2	Δ_2	Q_2	t_{r_2}
3	(x_3, y_3, z_3)	v_3	1	ϵ_3	d_3	Δ_3	Q_3	t_{r_3}
4	(x_4, y_4, z_4)	v_4	0	ϵ_4	d_4	Δ_4	Q_4	t_{r_4}

Figure 3.1 – Location information table (*Loc_tab*)

Algorithm 1 Updating the location information table.

Input: Location information msg(ID, (x,y,z), v, a, ϵ)
output: Updated location information table

1: Message reception time t_r = current time
2: Estimate distance d to the sender
3: Estimate the distance estimation error Δ
4: Estimate the link quality Q
5: Record the location information [ID, (x,y,z), v, a, ϵ, d, Δ, Q, t_r]

coordinates of the recorded location information, that is:

$$P(x, y, z) = \frac{\sum_{i=1}^{n} w_i \; L_i(x, y, z)}{\sum_{i=1}^{n} w_i} \tag{3.1}$$

$L_i(x, y, z)$ is the i^{th} recorded location information, w_i is the weight of $L_i(x, y, z)$ and n is the number of the recorded location information.

The weight w_i of the i^{th} location information entry is calculated as follows:

$$w_i = \frac{\alpha_1 w_{i1} + \alpha_2 w_{i2} + \alpha_3 w_{i3} + \alpha_4 w_{i4} + \alpha_5 w_{i5}}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5} \tag{3.2}$$

The weight w_{i1} depends on the source (sender) of the location information :

$$w_{i1} = \begin{cases} 1 & \text{if the sender is an anchor} \\ \frac{\epsilon}{1+\epsilon} & \text{if the sender is a normal node} \end{cases} \tag{3.3}$$

The location information collected from normal nodes is attributed lower weights than that received from anchor nodes. These weights depend on the location estimation quality ϵ.

The weight w_{i2} depends on the freshness of the stored location information:

$$w_{i2} = \frac{T}{T + \tau_i} \tag{3.4}$$

T is the maximum time that a received location information can be stored in the location information table; τ_i is the duration of time that the i^{th} location information has been stored.

The weight w_{i3} depends on the receiver/sender separating distance:

$$w_{i3} = \frac{\frac{1}{d_i+\Delta_i}}{\sum_{j=1}^{n} \frac{1}{d_j+\Delta_j}} \tag{3.5}$$

The variable d_i corresponds to the estimated distance separating the i^{th} state information sender and the normal node when the location message was sent; Δ_i describes the quality of the distance estimation. The higher is Δ_i the lower is the quality of the distance estimation. The variable Δ_i can be set to zero if the distance estimation error can't be evaluated.

The weight w_{i4} depends on the mobility of the sending node and is calculated as follows:

$$w_{i4} = \frac{v_{max}}{v_i + v_{max}} \tag{3.6}$$

The variable v_i corresponds to the velocity of the sender of the i^{th} recorded location information and v_{max} is the maximum velocity within the recorded velocities. The weight w_{i4} is particularly needed in networks with high propagation delays. In such networks, the sending node position may, depending on its velocity, has considerably changed when the message is received.

The weight w_{i5} depends on the link quality between the i^{th} location information sender and the receiver and is calculated as follows:

$$w_{i5} = \frac{Q_i}{Q_{max}} \tag{3.7}$$

The variable Q_i quantifies the link quality between the sender of the i^{th} recorded location information and the receiver, Q_{max} is the maximum link quality within the recorded link qualities. The RSSI can be used as a simple indicator of the link quality between two nodes.

The parameters α_1, α_2, α_3, α_4 and α_5 are set to either 1 or 0 depending on the availability of the corresponding information offering hence different configuration possibilities. For instance α_5 is set to 0 if there is no information about the link quality between the nodes and to 1 otherwise. The parameter α_3 is set to 0 if there is no information about the distance separating the sender and the receiver (range-free) and to 1 otherwise (range-based).

A normal node estimates the quality of its location estimation ϵ using the following

Algorithm 2 Location estimation.

Input: Location information table *Loc_tab*
output: node estimated position (x,y,z); estimation quality ϵ

1: $v_{max} = max(v_i)$; $i \in \{1, ..., n\}$
2: $Q_{max} = max(Q_i)$; $i \in \{1, ..., n\}$
3: **for** each stored location information i; $i \in \{1, ..., n\}$ **do**
4: τ_i = current time - t_r
5: Calculate w_{i1} using equation (3.3)
6: Calculate w_{i2} using equation (3.4)
7: Calculate w_{i3} using equation (3.5)
8: Calculate w_{i4} using equation (3.6)
9: Calculate w_{i5} using equation (3.7)
10: Calculate w_i using equation (3.2)
11: **end for**
12: Estimate my location (x,y,z) using equation (3.1)
13: Estimate my location estimation quality ϵ using equation (3.8)

equation:

$$\epsilon = \frac{\sum_{i=1}^{n} w_i}{n} \tag{3.8}$$

Algorithm 2 summarizes how a normal node estimates its position. First the node determines the maximum velocity v_{max} and the maximum link quality Q_{max} within all the recorded velocities and link qualities. Then, for each stored location information i, the normal node uses equations (3.3), (3.4), (3.5), (3.6), (3.7) and then (3.2) to estimate the weight of the considered location information entry. The normal node finally uses equations (3.1) and (3.8) to estimate its location and the location estimation quality ϵ.

3.2.2 Location information sharing

Algorithm 3 shows the behaviour of anchor and normal nodes. Anchor nodes periodically broadcast (each T_a) their location information. They also broadcast their location upon the reception of a location request message from a normal node. Anchors are not required to have the same fixed broadcast period T_a. Each anchor node may have its own adaptive broadcast period (e.g., depending on its remaining energy).

Normal nodes are continuously collecting location information messages sent from neighbouring nodes. If the localization is triggered (due to a given event or elapsed timer) at a given normal node u, then u uses the already collected location information to estimate its location. If this estimation does not satisfy a given requested quality ($\epsilon < \epsilon^*$) then, in order to ameliorate its estimate, node u broadcasts a location request message (*Req_msg*) including its identity and the quality of its current estimate. Normal nodes

Algorithm 3 Nodes behaviour.

Anchor Nodes

```
Periodically broadcast my position
if a location request message is received then
    Broadcast my position
end if
```

Normal Nodes

```
if a location information message is received then
    Update my location information table (algorithm 1)
end if
if a location request message (Req_msg (ID, ϵ_r) ) is received then
    Estimate my position and the estimation quality ϵ (algorithm 2)
    if ϵ > ϵ_r then
        send my position
    end if
end if
if localization is triggered then
    Estimate my position and the estimation quality ϵ (algorithm 2)
    if ϵ < ϵ* then
        Broadcast a location request message Req_msg (ID, ϵ)
        Collect the location request answers
        Estimate my position and the estimation quality ϵ (algorithm 2)
    end if
    if ϵ > ϵ** then
        Broadcast my position
    end if
end if
```

receiving such a request, estimate their location and the quality of their estimation and respond to this request only if their location estimate quality is better than the location estimate of the requester. Anchor nodes receiving the location request automatically respond by sending their position. The requesting node u collects the answers to its request and then estimates its location. Node u finally broadcasts its estimated position if it satisfies a given quality ($\epsilon > \epsilon^{**}$). Both ϵ^* and ϵ^{**} are parameters to fix. In particular, if $\epsilon^* = 0$ and $\epsilon^{**} = 1$ then only anchors location information will be used. For convenience, we summarize in table 3.1 the configurable parameters of our algorithm. We also provide some recommendations on the setting of these parameters.

We note that since normal nodes are continuously (and not at a precise time step) recording the received location announcements, anchor nodes are not required to be synchronized to send their location information at the same time step.

Parameter	Explanation
α_1, α_2, α_3, α_4, α_5	• Set to either 1 or 0 • Depend on the availability of the corresponding information: (link quality, nodes velocity, etc)
T	• Maximum time that a received location information can be stored in the location information table • Should be long enough to enable the nodes collect sufficient location announcements • But not very long so as not to have very old announcements
ϵ^*	• $0 \leq \epsilon^* \leq 1$ • Limit to send a location request message
ϵ^{**}	• $0 \leq \epsilon^{**} \leq 1$ • Limit to broadcasts an estimated position
T_a	• Anchors broadcast period • Each anchor may have its own adaptive broadcast period • A short T_a would improve the algorithm accuracy • But increase the communication overhead

Table 3.1 – ASAW configurable parameters

3.3 Performance Evaluation

In this section, we use simulations to evaluate the performance of our proposal. We also compare our results to those of two other localization techniques, namely the Centroid Localization (CL) [Bulu 00] and the Weighted Centroid Localization (WCL) [Blum 07] techniques.

3.3.1 Simulation setup

Our simulations were conducted using the OMNeT++ simulator [Varg 01]. For all our experiments, nodes evolve in a square area of $50m$ x $50m$. Fixed anchors are randomly placed using a uniform distribution. We use the random waypoint mobility model with fixed speed and no pause time [Roy 10] for the mobile anchors and real world human mobility traces [Rhee 09] for the normal nodes. The number of normal mobile nodes is fixed to $N = 19$. We adopt the simple path loss signal propagation model [Gold 05] under which the received signal strength P_r is expressed as:

$$P_r = \left(\frac{\lambda}{4\Pi}\right)^2 \cdot \left(\frac{1}{d}\right)^{\alpha} \cdot P_t \tag{3.9}$$

Where P_t is the maximum transmission power, d is the distance separating the sender and the receiver, α is the path loss coefficient, $\lambda = \frac{c}{f}$ is the wavelength of the transmitted

Parameter	Value
Area size	$50m$ x $50m$
Number of normal nodes	19
Maximum communication range R	$14m$
Simulation duration	1 hour
g	3
T_a	$3s$
T	$7s$
α_1, α_2, α_3, α_5	1
α_4	0
ϵ^*, ϵ^{**}	0.8

Table 3.2 – Simulation parameters

signal (c is the speed of light and f is the frequency of the transmitted signal). The path loss coefficient α depends on the propagation environment. In our simulations we set α to 4, which corresponds to a non-line-of-sight indoor environment [Rapp 96]. The maximum transmission power P_t is set to -34.1dbm resulting in a maximum transmission range of $14m$. The nodes communicate with each other using the IEEE 802.11 standard. The parameter g of the WCL algorithm is set to 3 as it is the most widely used value in literature [Blum 05]. The parameters of ASAW are set as follows: $T_a = 3s$, $T = 7s$, $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_5 = 1$, $\alpha_4 = 0$, $\epsilon^* = \epsilon^{**} = 0.8$. We use the RSSI as the link quality indicator (equation (3.7)). The distance separating a sending and a receiving node (equation (3.5)) is approximated through the RSSI received from the sending node. Each simulation scenario lasts 1 hour and was repeated 100 times (with different pseudorandom number generator seeds) in order to reach a confidence level of 95%. Table 3.2 summarizes the used simulation parameters.

The key metric for evaluating a localization technique is the accuracy of its location estimates. In this section, we study the accuracy of our algorithm by measuring its localization error. Localization error is the distance separating the actual position (x, y) and the estimated position $(\hat{x}, \hat{y})$ and is calculated as follows:

$$\sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$$

We assumed in our simulation that nodes are deployed in a two dimensional space. ASAW could, nevertheless, be equivalently used for three dimensional spaces.

3.3.2 Simulation results

We present next the obtained simulation results.

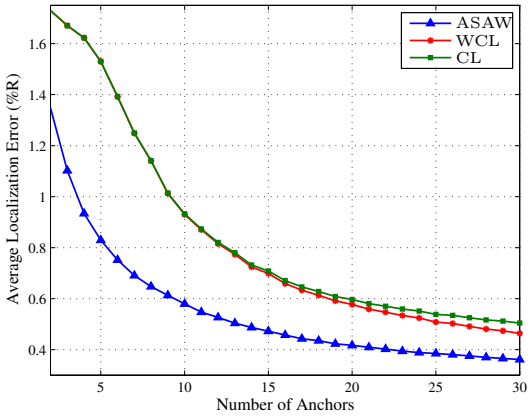
Comparison with other algorithms

Figure 3.2 shows the average localization error (i.e., the sum of the location errors of all normal nodes divided by the total number of normal nodes) obtained under ASAW, WCL and CL when the number of anchors increases from 2 to 30. Localization errors are normalized to the radio range R in order to allow an easy comparison with other state of the art localization techniques. We consider four different scenarios: (a) all the deployed anchor nodes are static, (b) all the deployed anchors are mobile with a velocity $v = 1m/s$ (c) all the deployed anchors are mobile with a velocity $v = 5m/s$ (d) a heterogeneous network were 50% of the deployed anchors are static and 50% are mobile with a velocity $v = 5m/s$.

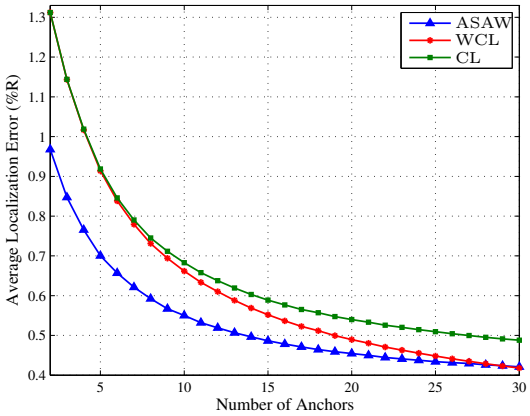
We can see that, for the three algorithms and under the four considered scenarios, the localization errors decrease as the number of anchors increases. This result is intuitive since the higher is the number of anchors the more location information will the normal nodes receive. Figure 3.2 also clearly shows that the location accuracy of our algorithm outperforms that of both WCL and CL. For instance with 2 anchors in the heterogeneous network scenario (Figure 3.2(d)), our algorithm provides an average location error of $0.79R$ in comparison to an average error of $1.68R$ for WCL and CL. The average location error decreases to $0.43R$ under ASAW, $0.52R$ under WCL and $0.55R$ under CL when the number of anchors increases to 16. Unlike WCL and CL that use the location information received at a given time step, our algorithm exploits the location information received within the hole time period T and attributes them weights depending on their freshness. This enables ASAW to provide much more accurate location estimations than WCL and CL particularly when the number of anchors is low (i.e., when the location information is scarce). The performance of CL and WCL are similar under a small number of anchors. For a large number of anchors, the accuracy of CL deteriorates in comparison with that of WCL. This is because, CL neglects the ranging information assuming that all the anchor nodes are equidistant from the unknown node. Nodes close and far from the true location are equivalently included in the averaging procedure, thereby corrupting the estimates.

Figure 3.3 compares the Cumulative Distribution Function (CDF) of the localization errors of the different algorithms. In Figure 3.3(a), we consider a network with 20 fixed anchors, (b) a network with 10 mobile anchors with a velocity $v = 1m/s$, (c) a network with 10 mobile anchors with a velocity $v = 5m/s$ (d) a heterogeneous network with 7 fixed anchors and 7 mobile anchors with a velocity $v = 5m/s$. We can see once again that ASAW has the best performance. Consider the 70th percentile in Figure 3.3(a), ASAW has a location error under $0.5R$ in comparison to $0.73R$ for WCL and 0.74 for CL.

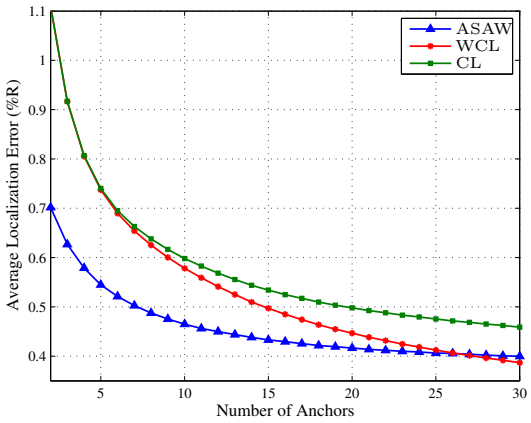
Figure 3.4 plots the localization error for a given normal node as a function of the simulation time. The considered scenario is of a network with 20 fixed anchors. As shown



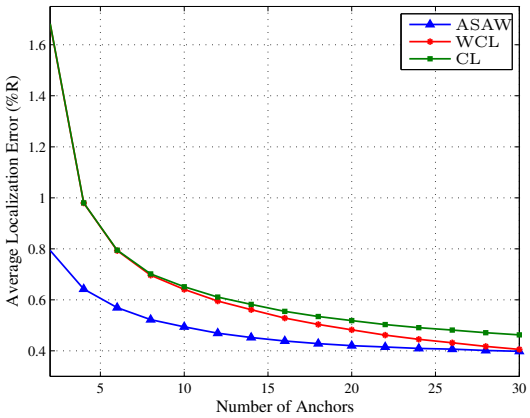
(a) Static anchors



(b) Mobile anchors ($v = 1m/s$)



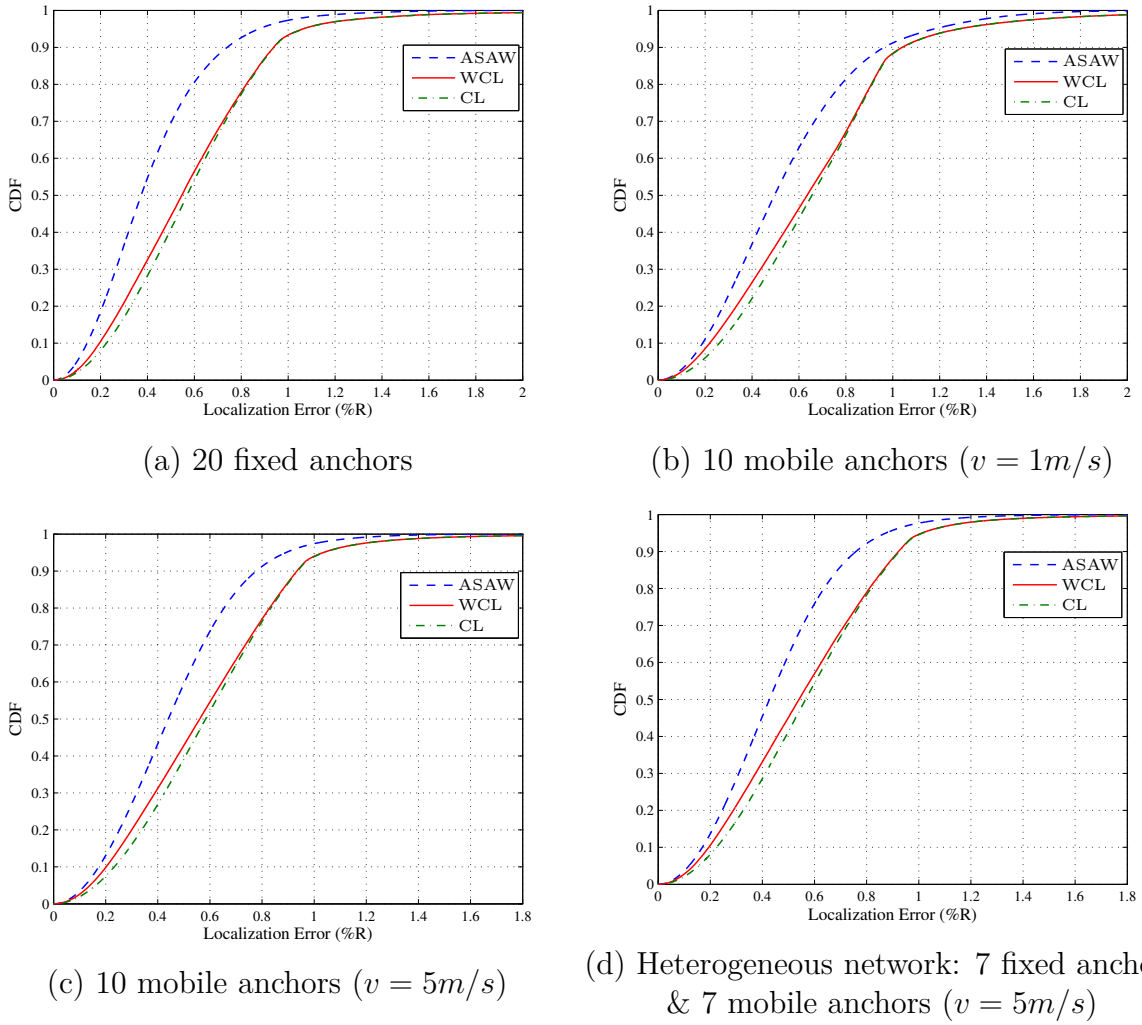
(c) Mobile anchors ($v = 5m/s$)



(d) Heterogeneous network: 50% static anchors & 50% mobile anchors ($v = 5m/s$)

Figure 3.2 – Localization error under different number of anchors

in the figure, ASAW provides the mobile node with the best location estimations during the hole simulation time. Taking a closer look at the performance of ASAW at Figure 3.4, we can notice some points where the localization errors are very small (close to zero, e.g., at times around 1150s indicated by the green arrow at the figure) and other points where the localization errors are the highest (around $1R$, e.g., at times around 2110s indicated by the red arrow at the figure). The points with the lowest location errors correspond to instants where the normal node is moving close to a sufficient number of anchors while the points with the highest localization errors correspond to instants where the normal node have no or a very low number of close anchors from which it can receive location announcements. Figure 3.5 illustrates the normal node actual positions within the deployment area at times around 1150s (blue crosses at the left of the figure) and at times around 2110s (blue crosses at the upper right side of the figure). The pink triangles represent the anchors positions. As shown in the figure, at times around 1150s,



(a) 20 fixed anchors

(b) 10 mobile anchors ($v = 1m/s$)

(c) 10 mobile anchors ($v = 5m/s$)

(d) Heterogeneous network: 7 fixed anchors & 7 mobile anchors ($v = 5m/s$)

Figure 3.3 – CDF of localization errors

the normal node is surrounded by an adequate number of close anchors and can hence accurately estimate its coordinates. At times around 2110s, the normal node is moving close to the border of the deployment area with no close-by anchors. In our simulations, anchor nodes were randomly placed within the deployment area. With such a non-optimal placement, a normal node may remain out of range of anchor nodes for a long time period. We believe that the use of an optimal anchor node placement algorithm [Ash 08] would ameliorate the location estimations. It is interesting to note that ASAW can estimate the normal nodes locations with a better accuracy than WCL and CL even though they remain out of range of anchors for a long time duration. This is because with ASAW, when a normal node have no or insufficient announcements from neighbouring anchors it asks its neighbouring normal nodes for their location estimations in order to ameliorate as possible its own estimations.

We summarize in table 3.3 the required number of anchors to reach an average lo-

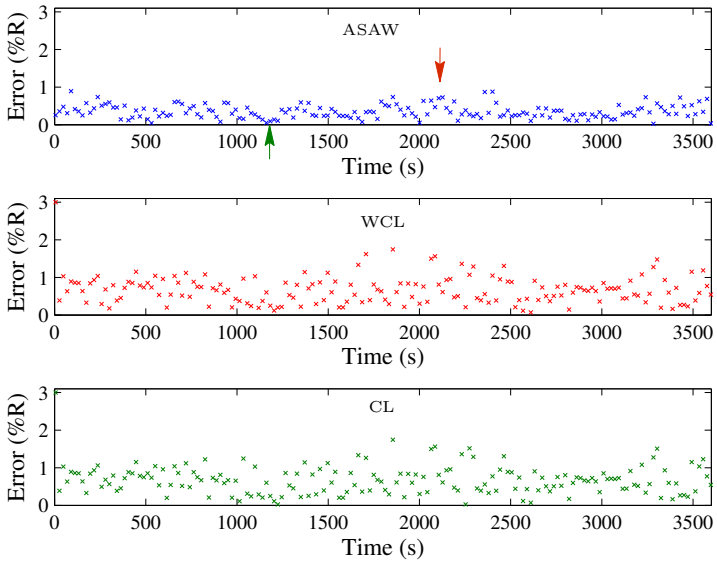


Figure 3.4 – Localization errors as a function of the simulation time

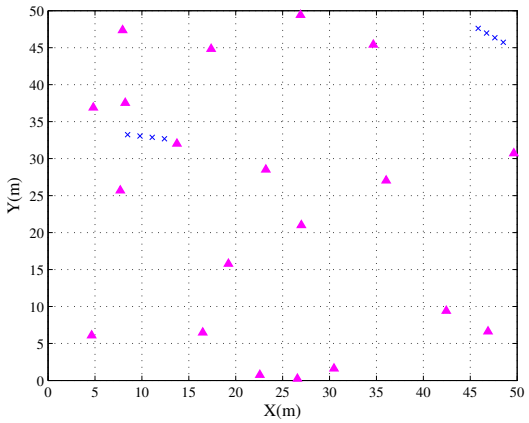


Figure 3.5 – Normal node traces

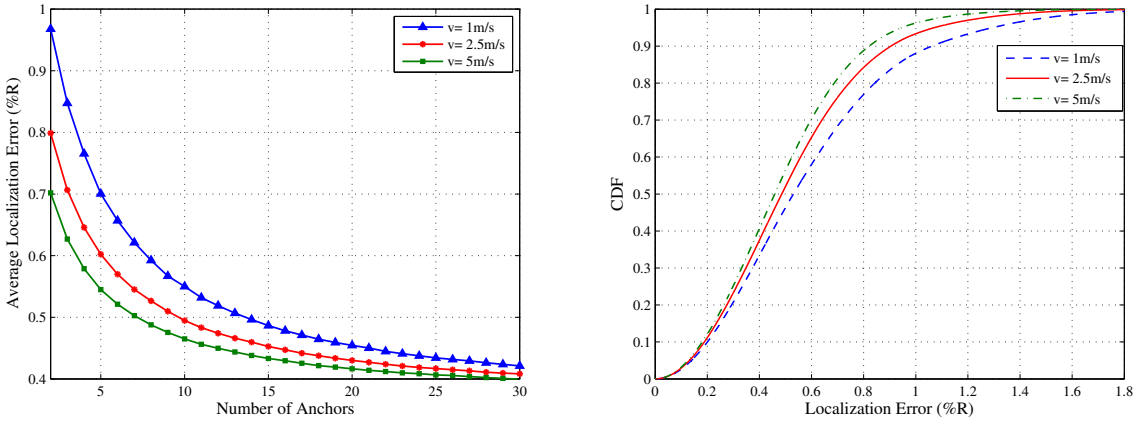
calization error less than or equal to a certain error bound Ω. We consider a scenario where all the anchor nodes are static. As expected, ASAW requires the lowest number of anchors. For example, to reach an average localization error less than or equal to $0.6R$, ASAW requires only 10 anchors while WCL requires 19 anchors and CL requires 20 anchors. Increasing the number of anchors makes the localization easier, but increases on the other hand the network and the deployment costs. Localization algorithms that require less anchors are thus preferred over those requiring a high anchors density.

Ω	$0.4R$	$0.5R$	$0.6R$	$0.7R$	$0.8R$
ASAW	22	13	10	7	6
WCL	> 30	27	19	15	13
CL	> 30	30	20	16	13

Table 3.3 – Required number of anchors to reach an error bound Ω

Impact of the anchors velocity

Figure 3.6 studies the impact of the anchors velocity on the localization error. Figure 3.6(a) plots the average localization error under three different velocities ($v = 1m/s$, $v = 2.5m/s$ and $v = 5m/s$) when the number of anchors increases from 2 to 30. Figure 3.6(b) plots the CDF of the localization errors under the three considered velocities with a number of anchors $n = 8$. We can see from this figure that the faster is the speed of the anchor nodes, the better is the location accuracy. Indeed, by moving quickly, anchor nodes will encounter and hence announce their locations to a higher number of normal nodes. For example, ASAW provides an average localization error of $0.64R$ when using 6 anchors moving with a velocity $v = 1m/s$. The average localization error decreases to $0.56R$ when the anchors velocity increases to $2.5m/s$ and to $0.50R$ when the anchors velocity increases to $5m/s$.



(a) Average localization error (b) CDF of localization errors

Figure 3.6 – Impact of the anchors velocity on the localization error

3.4 Conclusion

In this chapter, we presented ASAW a new distributed localization algorithm for mobile wireless networks. The proposed algorithm requires neither the synchronization between the nodes nor the prior knowledge of the nodes communication ranges. It besides weights

the collected location information by aggregating different metrics of different types providing hence a good robustness against the uncertainty of certain metrics. We considered different simulation scenarios in order to evaluate the performance of ASAW and compare it with that of the CL and the WCL algorithms. Simulations results showed that ASAW outperforms CL and WCL under all the considered scenarios. The study of our algorithm under different anchor nodes velocities has shown that the mobility can improve the accuracy.

Chapter 4

Spatio-temporal Conflict Detection and Alerting for Mobile Communicating Nodes

Contents

In this chapter we propose SLIDE a novel Straight LIne conflict DEtection and alerting algorithm for 3D-mobile communicating nodes. The main contributions of this chapter are manyfold and concern essentially:

- A distributed approach that requires little communication between the mobile nodes. The only information periodically exchanged between the nodes is their 3D position and velocity vectors. The mobile nodes intent information is not required.
- A comprehensive mathematical framework for 3D conflict detection where conflict conditions are stated in terms of simple inequalities and conflict parameters are expressed as a function of the current state information.
- A relaxation of the assumptions of perfect sensing and packet-loss free environment in order to improve the practicality of our proposal real world situations.
- An alerting algorithm based on our proposed analytical model.
- A validation of our proposed analytical model and alerting algorithm based on extensive simulations using OMNeT++.

The remaining of this chapter is organized as follows. Section 4.1 reviews previous works and points out their major drawbacks. Section 4.2 presents the mathematical basis of our algorithm. Section 4.3 describes SLIDE and provides practical recommendations on how to suitably choose its tuning parameters. Section 4.4 relaxes the assumptions of accurate state information and packet-loss free communications. Section 4.5 uses simulations to evaluate the performance of the proposed algorithm. Section 4.6 concludes the chapter.

4.1 3D Conflict Detection Algorithms

Because of its potential for dangerous impacts, conflict detection has become the focus of many active researches. Paper [Rebo 07] proposes a conflict detection approach for multiple aerial vehicles sharing a common airspace. The proposed algorithm assumes that each vehicle knows the precise planned 3D trajectories of all the other vehicles. The detection method is based on the discretization of the airspace: the airspace is divided into cubic cells and a conflict takes place if two vehicles lay in the same cell at the same time. This grid model simplifies the detection of conflicts since each vehicle needs only to check for the temporal overlapping between the cells of its trajectory and the cells of the other vehicles trajectories. The major drawback of this work consists in the fact that two vehicles that are not in the same cell may be closer than two other vehicles laying

in the same cell. This method cannot thus guarantee the separation requirement. The work in [Alej 13] resolves this drawback by defining the safety distance as a set of free cells between the vehicles. Yet, this work still assumes the availability of all the vehicles precise 3D trajectories. The author of [Belk 13] proposes a mathematical modeling and estimation of the conflict risk between a pair of vehicles. The conflict risk is expressed in terms of kinematic inequalities based on the time rates of both the bearing and the conflict cone angles. Such a formulation simplifies the conflict risks assessment since it does not explicitly require the use of the two vehicles speeds and orientations. The suggested approach was on the other hand limited to the determination of whether the two vehicles are on a conflict course. It provides no information about the upcoming conflict such as its point and time of occurrence. In [Cond 12] an aerial vehicle is represented with two joined boxes with a common centre. A horizontal box related to a minimum horizontal separation and a vertical box related to a vertical minimum separation. Each box is defined by three intervals, one interval on the x-axis, one on the y-axis and one on the z-axis. It is assumed that all the vehicles have known predetermined trajectories and a conflict is detected when there is an overlap between the intervals defining the boxes. The work in [Leve 11] proposes a conflict detection method that requires neither excessive processing nor complicated sensing mechanisms. Each mobile vehicle is assumed to be equipped with an on board radio modem. The communication range of the radio modem is defined as the radius of the safety zone and a conflict occurs once a communication link is established between two vehicles. The major problem of this method is that it is unable to detect future conflicts, it is limited to the detection of ongoing conflicts. In [Muno 13], a resolution advisory detection algorithm is presented. The proposed method is analogous to a conflict detection algorithm but predicts resolution advisories rather than loss of separation.

The existing conflict detection approaches are limited by two major drawbacks. First, they usually assume the availability of all the mobile nodes predetermined trajectories. Such an assumption may not be satisfied especially in highly dynamic environments (e.g., disaster zones) where trajectories have to be continuously adapted to the corresponding environment. Second, and more importantly, most of the proposed works assume perfect sensing capabilities and communication links enabling the nodes to collect precise information about their surrounding environment. This assumption does not hold in complex and harsh environments. Neglecting the environment uncertainties offers hence no guarantee of the non-underestimation of the conflict risks, resulting in violations of the safety requirements. These issues motivated us to propose the straight line conflict detection and alerting algorithm SLIDE. Indeed, with SLIDE nodes predetermined trajectories are no longer required. Our algorithm relies instead only on the periodically exchanged 3D

position and velocity vectors. Furthermore, SLIDE considers both sensing uncertainties and communication perturbations in order to guarantee its efficiency and applicability in real world situations.

4.2 3D Conflict Detection

In this section, we propose a novel mathematical framework for the 3D conflict detection. We first present the considered mathematical notations and then derive the conflict conditions and parameters. We introduce assumption A(1) that is useful for our initial analysis. This assumption will be relaxed later on in section 4.4.

A (1). *Each mobile node knows its accurate 3D position and velocity vectors.*

4.2.1 Mathematical notations

We consider a 3 dimensional airspace with two distinguished mobile nodes A and B. The following 3D vectors are used to represent the two mobile nodes accurate state information at time $t = 0$:

$$\mathbf{P}_a,\ \mathbf{P}_b : \text{nodes } A \text{ and } B \text{ positions}$$
$$\mathbf{V}_a,\ \mathbf{V}_b: \text{nodes } A \text{ and } B \text{ velocities}$$

Nodes dynamics are represented by a point linearly moving at a constant speed. Thus, the predicted positions of nodes A and B at any time t are given respectively by:

$$\mathbf{P}_a + t\mathbf{V}_a \text{ and } \mathbf{P}_b + t\mathbf{V}_b$$

We note:

$$\mathbf{P}_\Delta = \mathbf{P}_a \text{ - } \mathbf{P}_b\text{: the relative position vector}$$
$$\mathbf{V}_\Delta = \mathbf{V}_a \text{ - } \mathbf{V}_b\text{: the relative velocity vector}$$

At any time t, the distance separating the two nodes can be expressed in terms of the relative position and velocity vectors as follows:

$$d(t) = \|\mathbf{P}_\Delta + t\mathbf{V}_\Delta\|$$

The notation $\|\mathbf{U}\|$ stands for the norm of $\mathbf{U}(u_x,u_y,u_z)$, that is $\|\mathbf{U}\| = \sqrt{u_x^2 + u_y^2 + u_z^2}$. Hereafter, the notation $\mathbf{U}.\mathbf{U}'$ will be used to refer to the dot product of vectors $\mathbf{U}$ and $\mathbf{U}'$. For the sake of clarity and simplicity of notations, relative position and velocity vectors $\mathbf{V}_\Delta$ and $\mathbf{P}_\Delta$ will be used instead of the individual vectors $\mathbf{P}_a$, $\mathbf{P}_b$,$\mathbf{V}_a$, $\mathbf{V}_b$.

For convenience, we summarize in table 4.1 the important mathematical notations used throughout this chapter.

Notation	Meaning
$\mathbf{P}_a$, $\mathbf{P}_b$	Nodes A and B certain positions
$\mathbf{V}_a$, $\mathbf{V}_b$	Nodes A and B certain velocities
$\mathbf{P}_a^u$, $\mathbf{P}_b^u$	Nodes A and B uncertain positions
$\mathbf{V}_a^u$, $\mathbf{V}_b^u$	Nodes A and B uncertain velocities
$\mathbf{P}_\Delta$	Certain relative position
$\mathbf{V}_\Delta$	Certain relative velocity
$\mathbf{P}_\Delta^u$	Uncertain relative position
$\mathbf{V}_\Delta^u$	Uncertain relative velocity
T_l	Look ahead time
R_a, R_b	Nodes A and B protected zones Radii
t_{ca}	Time of closest approach
d_{ca}	Distance of closest approach
$d(A, B)$	Distance separating the Nodes A and B
t_{in}	Time of loss of separation
t_{out}	Time of the end of loss of separation
T_b	Broadcast cycle
R_c	Communication range
ρ_a, ρ_b	Nodes A and B positions uncertainties bounds
υ_a, υ_b	Nodes A and B velocities uncertainties bounds
R_{unc}	Uncertainty distance
p	Packet loss probability

Table 4.1 – Notations list

4.2.2 Conflict conditions and parameters

For a safe navigation of the mobile nodes in a 3 dimensional space, we assume that each mobile node is surrounded by a virtual spherical protected zone that should not overlap with another node protected zone. A predicted conflict between nodes A and B occurs when there is a future time t within a look ahead time T_l at which the protected zones of the two nodes overlap. The look ahead time T_l is a prediction time, it describes how far in advance future conflicts can be detected.

Definition 4 (Conflict)**.** *mobile nodes A and B are in conflict if:*

$$\exists\ t \in [0, T_l] /\ d(t) \leq R_a + R_b$$

Where R_a and R_b are the radii of the nodes spherical protected zones.

Definition 5 (Time of closest approach t_{ca})**.** *The time of closest approach between the mobile nodes A and B is the instance at which the two nodes reach their minimum separation distance (Figure 4.1).*

Definition 6 (Distance of closest approach d_{ca})**.** *The distance of closest approach between nodes A and B is their separating distance at the time of closest approach* t_{ca}*.*

Proposition 1. *The time of closest approach* t_{ca} *is given by:*

$$t_{ca} = \begin{cases} 0 & if \ \ \boldsymbol{V}_\Delta = 0 \\ \frac{-(\boldsymbol{P}_\Delta . \boldsymbol{V}_\Delta)}{\boldsymbol{V}_\Delta{}^2} & if \ \ \boldsymbol{V}_\Delta \neq 0 \end{cases}$$

Proof. t_{ca} is the time that minimizes the separating distance between the mobile nodes A and B. Since the distance between two points cannot be negative, $d(t)$ reaches its minimum when $d(t)^2$ achieves its minimum. The minimum consequently occurs when the derivative of:

$$d(t)^2 = \|\mathbf{P}_\Delta + t\mathbf{V}_\Delta\|^2 = \mathbf{V}_\Delta{}^2 \ t^2 + 2(\mathbf{P}_\Delta . \mathbf{V}_\Delta) \ t + \mathbf{P}_\Delta{}^2$$

is equal to zero:

$$\frac{d(d(t)^2)}{dt} = 2\mathbf{V}_\Delta{}^2 \ t + 2(\mathbf{P}_\Delta . \mathbf{V}_\Delta) = 0 \tag{4.1}$$

If $\mathbf{V}_\Delta \neq 0$, solving equation (4.1) for t readily gives: $t_{ca} = \frac{-(\mathbf{P}_\Delta . \mathbf{V}_\Delta)}{\mathbf{V}_\Delta{}^2}$. If $\mathbf{V}_\Delta = 0$ then the two nodes are moving in parallel and will remain apart at the same distance. The time of closest approach t_{ca} can consequently be set to zero.
The distance at the closest approach d_{ca} is found by calculating $d(t_{ca})$:

$$d_{ca} = d(t_{ca}) = \|\mathbf{P}_\Delta + t_{ca}\mathbf{V}_\Delta\| \tag{4.2}$$

□

Definition 7 (Time of loss of separation t_{in})**.** *The time of loss of separation between the mobile nodes A and B is the instant at which starts the overlapping of the two nodes protected zones (Figure 4.1).*

Definition 8 (Time of the end of loss of separation t_{out})**.** *The time of the end of loss of separation between the mobile nodes A and B is the instant at which ends the overlapping of the two nodes protected zones (Figure 4.1).*

Proposition 2. *If* $(\boldsymbol{P}_\Delta . \boldsymbol{V}_\Delta)^2 - \boldsymbol{V}_\Delta{}^2(\boldsymbol{P}_\Delta{}^2 - (R_a + R_b)^2) \geq 0$ *and* $\boldsymbol{V}_\Delta \neq 0$*,* t_{in} *and* t_{out} *are given by:*

$$t_{in} = \frac{-(\boldsymbol{P}_\Delta . \boldsymbol{V}_\Delta) - \sqrt{(\boldsymbol{P}_\Delta . \boldsymbol{V}_\Delta)^2 - \boldsymbol{V}_\Delta{}^2(\boldsymbol{P}_\Delta{}^2 - (R_a + R_b)^2)}}{\boldsymbol{V}_\Delta{}^2} \tag{4.3}$$

$$t_{out} = \frac{-(\boldsymbol{P}_\Delta.\boldsymbol{V}_\Delta) + \sqrt{(\boldsymbol{P}_\Delta.\boldsymbol{V}_\Delta)^2 - \boldsymbol{V}_\Delta{}^2(\boldsymbol{P}_\Delta{}^2 - (R_a + R_b)^2)}}{\boldsymbol{V}_\Delta{}^2}$$

Proof. t_{in} and t_{out} are the instants at which the two mobile nodes separating distance gets equal to $R_a + R_b$:

$$\|\mathbf{P}_\Delta + t\mathbf{V}_\Delta\| = R_a + R_b \tag{4.4}$$

squaring equation (4.4) gives:

$$\mathbf{V}_\Delta{}^2\ t^2 + 2(\mathbf{P}_\Delta.\mathbf{V}_\Delta)\ t + \mathbf{P}_\Delta{}^2 - (R_a + R_b)^2 = 0 \tag{4.5}$$

Equation (4.5) is quadratic in t with $a = \mathbf{V}_\Delta{}^2$, $b = (\mathbf{P}_\Delta.\mathbf{V}_\Delta)$ and $c = \mathbf{P}_\Delta{}^2 - (R_a + R_b)^2$. If $b^2 - ac \geq 0$ and $a \neq 0$, equation (4.5) accepts two solutions:

$$t_{in} = \frac{-b - \sqrt{b^2 - ac}}{a}$$
$$t_{out} = \frac{-b + \sqrt{b^2 - ac}}{a}$$

The time t_{in} represents the time of the beginning of the conflict, t_{out} the time of the end of the conflict and the interval $[t_{in}, t_{out}]$ represents the conflict interval (Figure 4.1). □

Proposition 3. *If $\boldsymbol{V}_\Delta \neq 0$, we have:*

$$d_{ca} \leq (R_a + R_b) \iff$$

$$(\boldsymbol{P}_\Delta.\boldsymbol{V}_\Delta)^2 - \boldsymbol{V}_\Delta{}^2(\boldsymbol{P}_\Delta{}^2 - (R_a + R_b)^2) \geq 0$$

Proof.

$$d_{ca} = \|\mathbf{P}_\Delta + t_{ca}\ \mathbf{V}_\Delta\| \leq (R_a + R_b)$$
$$\Leftrightarrow \|\mathbf{P}_\Delta + t_{ca}\ \mathbf{V}_\Delta\|^2 \leq (R_a + R_b)^2$$
$$\Leftrightarrow \mathbf{P}_\Delta{}^2 + t_{ca}^2\ \mathbf{V}_\Delta{}^2 + 2t_{ca}\ (\mathbf{P}_\Delta.\mathbf{V}_\Delta) \leq (R_a + R_b)^2$$

substituting t_{ca} by $\frac{-(\mathbf{P}_\Delta.\mathbf{V}_\Delta)}{\mathbf{V}_\Delta{}^2}$ we have:

$$\mathbf{P}_\Delta{}^2 - \frac{(\mathbf{P}_\Delta.\mathbf{V}_\Delta)^2}{\mathbf{V}_\Delta{}^2} \leq (R_a + R_b)^2$$
$$\Leftrightarrow \frac{(\mathbf{P}_\Delta.\mathbf{V}_\Delta)^2}{\mathbf{V}_\Delta{}^2} - \mathbf{P}_\Delta{}^2 + (R_a + R_b)^2 \geq 0$$

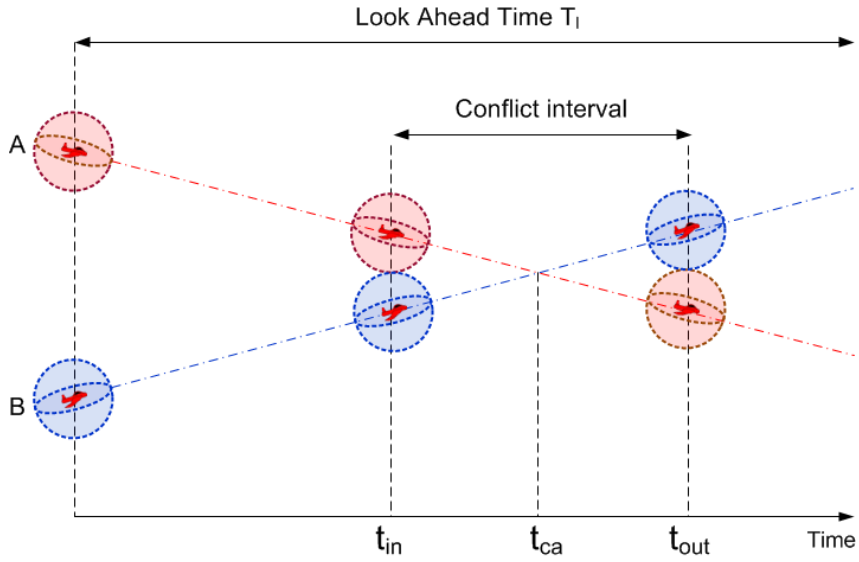


Figure 4.1 – Conflict parameters

multiplying both sides by ${\mathbf{V}_\Delta}^2$ we have:

$$(\mathbf{P}_\Delta.\mathbf{V}_\Delta)^2 - {\mathbf{V}_\Delta}^2({\mathbf{P}_\Delta}^2 - (R_a + R_b)^2) \geq 0$$

□

Proposition 3 implies that if $d_{ca} \leq (R_a + R_b)$ and $\mathbf{V}_\Delta \neq 0$ then the expression under the square root in equation (4.3) is positive and hence t_{in} exists and can be calculated using this equation.

4.3 SLIDE: a Straight Line Conflict Detection and Alerting Algorithm

In this section, we propose SLIDE a novel straight line conflict detection and alerting algorithm. We first describe the proposed algorithm with a straightforward pseudo-code and then give some recommendations on how to suitably set the values of its tuning parameters. SLIDE has particularly two tuning parameters which are the broadcast cycle of the state information T_b and the look ahead time T_l.

4.3.1 SLIDE definition

The proposed algorithm is distributed and state based: each mobile node individually uses its own state information and that of neighbouring nodes to detect future conflicts.

Algorithm 4 presents the pseudo-code of SLIDE. The algorithm takes as parameters the broadcast cycle T_b and the look ahead time T_l and returns whether the mobile node will encounter a loss of separation within T_l. Each mobile node periodically (each T_b) broadcasts a STATE message with its current position vector, velocity vector and protected zone radius. A back-off time is used in order to avoid packet collisions among multiple nodes simultaneously sending their STATE messages. Before sending its first STATE message, each node sets a back-off time of τ seconds, where τ is uniformly distributed in $[0, T_{off}]$. The maximal back-off time T_{off} should be set according to the network density. In particular, in dense networks the value of T_{off} should be increased in order to prevent simultaneous broadcasts.

If a mobile node receives a STATE message from a neighbouring node then:

- It estimates t_{ca} from proposition 1.
- If $t_{ca} < 0$, then the two mobile nodes are traveling away from each other and no future conflict is predicted to happen (the two nodes pathways extrapolated backward lead to a minimum separation distance).
- If $t_{ca} \geq 0$, then there is a time at which the two mobile nodes reach their closest point of approach. The separating distance at the closest approach d_{ca} can consequently be calculated using equation (4.2).
- If $d_{ca} > (R_a + R_b)$ then no future conflict is predicted.
- If $d_{ca} \leq (R_a + R_b)$, then a conflict is predicted to happen at t_{in}:
  - If $\mathbf{V}_\Delta = 0$ then the two mobile nodes will remain at the same current separating distance and the time of loss of separation t_{in} can hence be set to zero.
  - If $\mathbf{V} \neq 0$ then t_{in} can be calculated using equation (4.3) . Indeed, we have here $d_{ca} \leq (R_a + R_b)$ and $\mathbf{V} \neq 0$ and consequently according to proposition 3, t_{in} can be calculated using equation (4.3).
- If $0 \leq t_{in} \leq T_l$, then the mobile node first registers the conflict and then raises a conflict alert.

Each mobile node maintains a conflict table in which it registers the predicted conflicts information such as the conflicting nodes identities, predicted t_{in} and t_{out}. The node identifies simultaneous conflicts with multiple other nodes by searching the temporal overlapping of the registered conflicts intervals.

Figure 4.2 and Figure 4.3 represent examples of encounter scenarios between a pair of mobile nodes. Figure 4.2 shows a head on encounter scenario. Node B (the encounter

Algorithm 4 3D conflict detection algorithm.

Input: T_l, T_b

1: Broadcast each T_b current state_msg($\mathbf{P}_a$,$\mathbf{V}_a$,R_a)
2: **if** state_msg($\mathbf{P}_b$,$\mathbf{V}_b$,R_b) received **then**
3: $\mathbf{P}_\Delta = \mathbf{P}_a - \mathbf{P}_b$
4: $\mathbf{V}_\Delta = \mathbf{V}_a - \mathbf{V}_b$
5: **if** $\mathbf{V}_\Delta = 0$ **then**
6: $t_{ca} = 0$
7: **else**
8: $t_{ca} = \frac{-(\mathbf{P}_\Delta.\mathbf{V}_\Delta)}{{\mathbf{V}_\Delta}^2}$
9: **end if**
10: **if** $t_{ca} < 0$ **then**
11: **return** -1
12: **else**
13: $d_{ca} = \|\mathbf{P}_\Delta + t_{ca}\mathbf{V}_\Delta\|$
14: **end if**
15: **if** $d_{ca} > (R_a + R_b)$ **then**
16: **return** -1
17: **else**
18: **if** $\mathbf{V}_\Delta = 0$ **then**
19: $t_{in} = 0$
20: **else**
21: $a = {\mathbf{V}_\Delta}^2$
22: $b = (\mathbf{P}_\Delta.\mathbf{V})$
23: $c = {\mathbf{P}_\Delta}^2 - (R_a + R_b)^2$
24: $t_{in} = \frac{-b-\sqrt{b^2-ac}}{a}$
25: **end if**
26: **end if**
27: **if** $0 \leq t_{in} \leq T_l$ **then**
28: register conflict
29: **return** 1
30: **else**
31: **return** -1
32: **end if**
33: **end if**

node) broadcasts a STATE message containing its current position vector $\mathbf{P}_b$, velocity vector $\mathbf{V}_b$ and protected zone radius R_b. Upon receiving this information, node A linearly projects its current state information as well as the encounter state information into the near future in order to derive the conflict parameters and decide upon the conflict risk

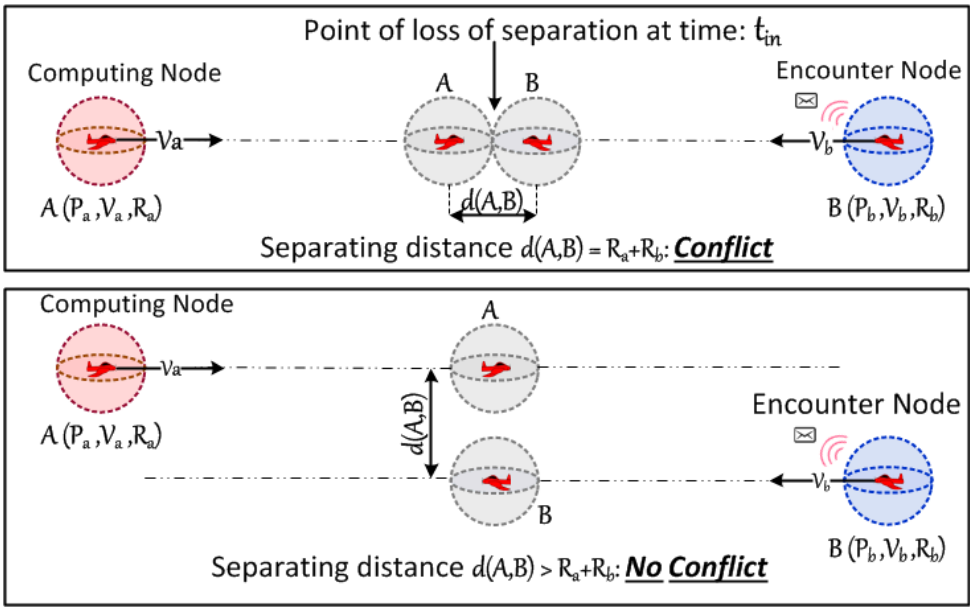


Figure 4.2 – Head on encounter scenario

with the encounter. In the first case (first upper subfigure of Figure 4.2), node A detects a future loss of separation at time t_{in} and will therefore register the detected future conflict and raise a conflict alert. In the second case (second subfigure of Figure 4.2), the two mobile nodes will remain in the near future at the same separating distance $d(A, B) > R_a + R_b$. There is thus no near future conflict risk. Figure 4.3 shows a crossing encounter scenario. Similarly, upon receiving a STATE message from the encounter mobile node, node A linearly projects the state information into the near future and then estimates the conflict risk. In the first case (first upper subfigure of Figure 4.3), a conflict is predicted to take place at time t_{in}. However in the second case (second subfigure of Figure 4.3), node A is faster then node B. It will pass and get away from the two trajectories intersection point while node B is still far away from the intersection point. As such, no conflict is predicted to take place.

4.3.2 Setting of the tuning parameters

For an efficient conflict detection, the values of the tuning parameters T_b and T_l have to be suitably chosen.

Setting of the broadcast cycle T_b

The broadcast cycle T_b strongly affects the number of missed alarms (i.e., conflicts with no prior issued alerts). If T_b is too large, the mobile nodes will not have enough information about their surrounding traffic and consequently several conflicts may occur without being

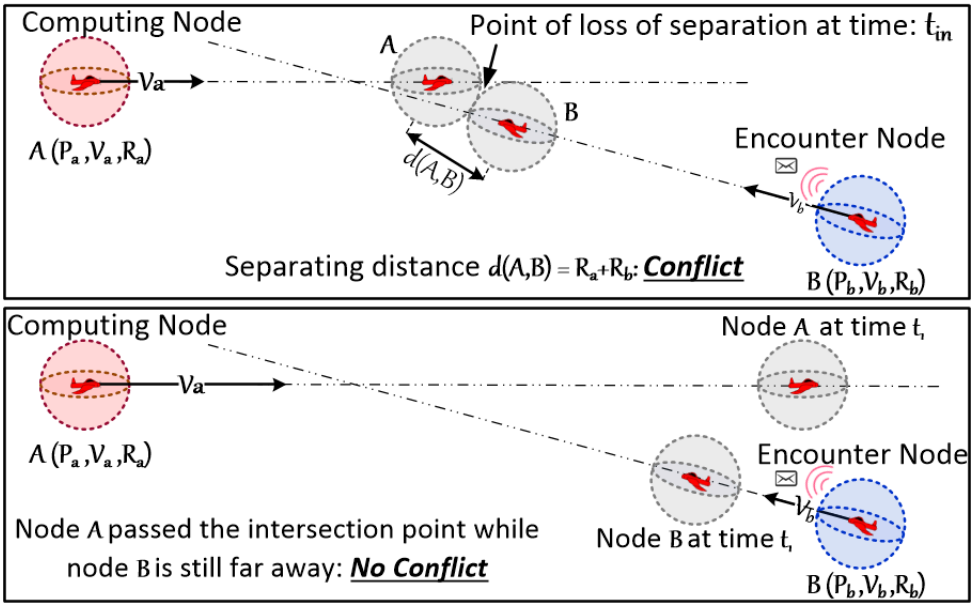


Figure 4.3 – Crossing encounter scenario

detected. Small values of T_b would reduce the number of missed alarms but at the expense of redundant information and repetitive calculations. Next, we introduce the following assumption:

A (2). *There is no packet loss in the network.*

Assumption A(2) guarantees that each mobile node can successfully transmit its STATE messages to its neighbouring nodes without any loss. In section 4.4, we will relax this assumption by considering losses in the STATE messages.

We consider two distinguished nodes linearly moving, in a sufficiently large area, at constant speeds v_a and v_b and surrounded by protected zones of radii R_a and R_b respectively. Nodes A and B can communicate with each other if their separating distance $d(A, B)$ is *strictly* less than the communication range R_c. We would like to find the appropriate broadcast cycle T_b that guarantees no missed alarms. To this end, let us consider the worst case scenario where at instant t_1 we have, as portrayed on the upper subfigure of Figure 4.4, (1) the two mobile nodes are aligned in a head on situation (2) $d(A, B)_{t_1} = R_c$ (3) node A (or equivalently node B) broadcasts a STATE message (note that this message will not be received by the other mobile node since $d(A, B)_{t_1}$ is not *strictly* less than the communication range R_c). The broadcast cycle T_b should be less than the time needed for the two nodes A and B to get in contact. As such, and in order to avoid missing conflicts within T_b, we should have:

$$d(A, B)_{t_1+T_b} \geq R_a + R_b$$

$$\Leftrightarrow d(A,B)_{t_1} - (v_a T_b + v_b T_b) \geq R_a + R_b$$

$$\Leftrightarrow R_c - T_b(v_a + v_b) \geq R_a + R_b$$

$$\Leftrightarrow T_b \leq \frac{R_c - (R_a + R_b)}{v_a + v_b} \tag{4.6}$$

Figure 4.4 represents an example of the considered worst case scenario with $v_a = v_b = 1m/s$; $R_a = R_b = 2m$ and $R_c = 10m$. According to equation (4.6), to avoid missed alarms T_b should verify: $T_b \leq 3s$. With a broadcast cycle $T_b \leq 3s$, for example $T_b = 1.5s$ (as shown in case (1) in Figure 4.4), the STATE message will be sent while $d(A,B) > R_a + R_b$ and consequently the conflict will be detected before its occurrence. With a broadcast cycle $T_b = 3s$ (case (2) in Figure 4.4), the STATE message will be sent while $d(A,B) = R_a + R_b$ and consequently the conflict will be detected just at the instant of the beginning of the loss of separation. With a broadcast cycle $T_b > 3s$, for example $T_b = 4s$ (case (3) in Figure 4.4), the STATE message will be sent while $d(A,B) < R_a + R_b$ and consequently the conflict will be detected very late (missed). Finally, with a broadcast cycle largely greater than $3s$, for example $T_b = 12s$ (case (4) in Figure 4.4), the STATE message will be sent while the two mobile nodes are no longer within the communication range of each other ($d(A,B) > R_c$). The STATE message will not be received and the conflict will not be detected.

If the two nodes have two different communication ranges R_{ca} and R_{cb} then T_b should verify:

$$T_b \leq \frac{min(R_{ca}, R_{cb}) - (R_a + R_b)}{v_a + v_b} \tag{4.7}$$

Equation (4.7) can be intuitively explained by the fact that the higher are the velocities and the protected zone radii, the shorter should be T_b; and the larger are the nodes communication ranges, the more state information they have on their surrounding environment and consequently the higher can T_b be set.

Equation (4.7) may be further exploited to derive a lower bound for the communication ranges and an upper bound for the velocities and the protected zone radii. For example, for a given broadcast cycle T_b, velocities v_a and v_b, protected zone radii R_a and R_b, the communication ranges should be set as follows in order to avoid missed alarms:

$$min(R_{ca}, R_{cb}) \geq T_b(v_a + v_b) + R_a + R_b \tag{4.8}$$

Similarly, for a given broadcast cycle T_b, velocities v_a and v_b, communication ranges R_{ca} and R_{cb}, the two nodes protected zones should satisfy:

$$R_a + R_b \leq min(R_{ca}, R_{cb}) - T_b(v_a + v_b) \tag{4.9}$$

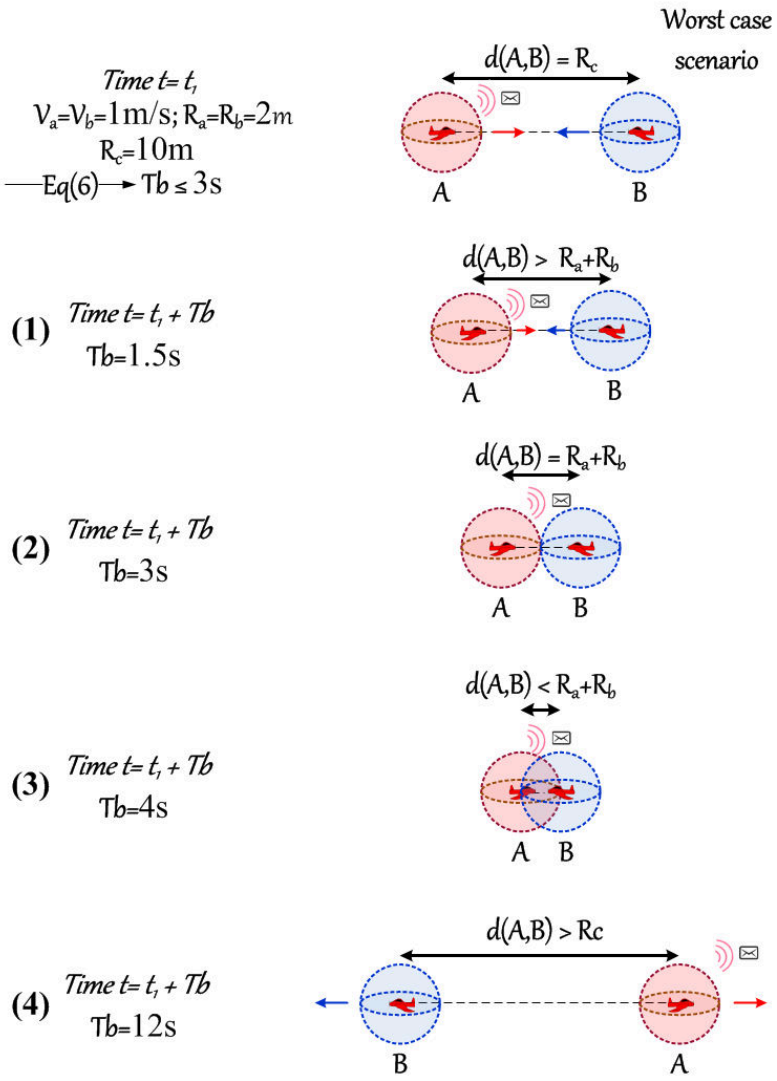


Figure 4.4 – Broadcast cycle T_b setting: worst case scenario

Setting of the look ahead time T_l

With a very large prediction time T_l, the accuracy of the straight line trajectory prediction would be inappropriate which leads to an increased number of false alarms (i.e., issued alerts without a subsequent conflicts). Indeed, straight line projection does not take into account the fact that mobile nodes may change their directions, an aspect that is very probable in a long propagation time. Figure 4.5 shows an example of a falsely predicted conflict. The conflict detection algorithm considered that the two mobile nodes A and B will have a straight trajectory throughout the look ahead time T_l and consequently predicted a conflict at time t_2. However, at time t_1 ($t_1 < t_2$), node B has changed its direction and the predicted conflict had not actually took place. Small values of T_l would decrease false alarms but on the other hand, only close conflicts could be detected giving

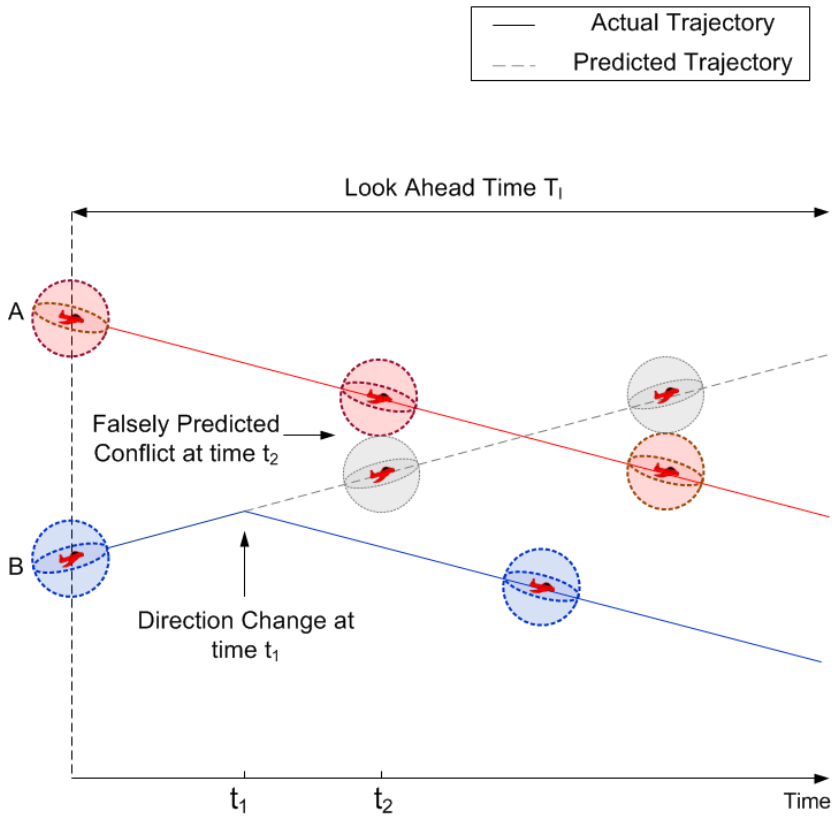


Figure 4.5 – Example of a falsely predicted conflict

little time for avoidance manoeuvres. Typically, the choice of T_l should be set according to the mobility of the nodes. If the nodes have frequent direction changes then T_l should be accordingly short and vice versa.

4.4 Packet Loss and State Information Errors

We have previously assumed that each mobile node knows its exact state information (i.e., assumption A(1)) and that it can successfully transmit this information to its surrounding nodes without any loss (i.e., assumption A(2)). Such assumptions may not hold in the real world. Indeed, existing location estimation techniques are not very accurate [Wang 10] and real world wireless networks are prone to packet losses [Bacc 12] that may negatively impact the information collected about neighbours. In this section we discuss how to relax these two assumptions.

4.4.1 Packet loss

The use of the back-off mechanism avoids the possibility of multiple neighbouring nodes sending their STATE messages at the same time and hence reduces the probability of

STATE messages loss due to collisions. Packet loss may nevertheless occur due to other supplementary reasons such as signal attenuation, misconfiguration and malicious drops [Ning 12].

We consider:

p: the probability of failure on each transmitted message. The probability p can be measured using empirical or theoretical packet loss estimation approaches [Halp 11, Jaco 15].

N: the number of transmission trials till the first successful transmission.

N follows a geometric distribution with a cumulative distribution function $F(n) = P(N \leq n) = 1-p^n$. The inverse distribution function of N is $F^{-1}(u) = \frac{\log(1-u)}{\log(p)}$, $0 < u < 1$. For a given p, we have $N \leq F^{-1}(u)$ with a probability u. For example with a high loss probability $p = 0.4$ and $u = 0.99$, we have $F^{-1}(u) = 5$, which means that 99% of the messages are successfully transmitted within the 5^{th} trial. With a low loss probability $p = 0.01$ and $u = 0.99$, we have $F^{-1}(u) = 1$, that is 99% of the messages are successfully transmitted within the 1^{st} attempt.

To counteract the packet loss, we hence propose to use a new broadcast cycle T_b^* that enables the transmission of $F^{-1}(u)$ (with u close to 1) messages within the initial broadcast cycle T_b, that is:

$$T_b^* = \frac{T_b}{F^{-1}(u)} = \frac{\log(p)}{\log(1-u)} \times T_b \tag{4.10}$$

Recall that T_b is the required broadcast cycle in a packet-loss free environment (equation (4.7)). From this analysis and equation (4.7), we can state the following proposition:

Proposition 4. *: In a packet-loss prone environment, to avoid missing conflicts between two nodes A and B linearly moving at constant speeds, the STATE messages broadcast cycle T_b^* should verify.*

$$T_b^* \leq \frac{\log(p)}{\log(1-u)} \times \frac{min(R_{ca}, R_{cb}) - (R_a + R_b)}{v_a + v_b} \tag{4.11}$$

Where p is the packet loss probability, u is a probability close to 1, v_a and v_b are the two nodes velocities, R_{ca} and R_{cb} are the two nodes communication ranges and R_a and R_b are the two nodes protected zones radii.

4.4.2 State information errors

To detect future conflicts, each mobile node requires its own and its neighbouring nodes current positions and velocities. This state information inevitably contains uncertainties because of sensors imperfection. Next, we consider uncertainties in the measurement of the mobile nodes positions and velocities. That is, the exact state information is no longer available; nodes are only aware of the uncertain state information. We use the following 3D vectors to represent two distinguished mobile nodes A and B certain and uncertain state information:

$$\mathbf{P}_a, \mathbf{P}_b : \text{certain positions}$$
$$\mathbf{V}_a, \mathbf{V}_b : \text{certain velocities}$$
$$\mathbf{P}^u_a, \mathbf{P}^u_b : \text{uncertain positions}$$
$$\mathbf{V}^u_a, \mathbf{V}^u_b : \text{uncertain velocities}$$
$$\mathbf{P}_\Delta = \mathbf{P}_a - \mathbf{P}_b : \text{certain relative position}$$
$$\mathbf{V}_\Delta = \mathbf{V}_a - \mathbf{V}_b : \text{certain relative velocity}$$
$$\mathbf{P}^u_\Delta = \mathbf{P}^u_a - \mathbf{P}^u_b : \text{uncertain relative position}$$
$$\mathbf{V}^u_\Delta = \mathbf{V}^u_a - \mathbf{V}^u_b : \text{uncertain relative velocity}$$

We assume that bounds on the state information uncertainties are known:

$$\|\mathbf{P}_a - \mathbf{P}^u_a\| \leq \rho_a$$
$$\|\mathbf{P}_b - \mathbf{P}^u_b\| \leq \rho_b$$
$$\|\mathbf{V}_a - \mathbf{V}^u_a\| \leq \upsilon_a$$
$$\|\mathbf{V}_b - \mathbf{V}^u_b\| \leq \upsilon_b$$

Where ρ_a and ρ_b are the two mobile nodes position uncertainties bounds; υ_a and υ_b are the two mobile nodes velocities uncertainties bounds.

Proposition 5. *The time of closest approach t_{ca} can be upper bounded by the uncertain state information as follows:*

$$t_{ca} \leq \frac{\|\boldsymbol{P}^u_\Delta\| + (\rho_a + \rho_b)}{\|\boldsymbol{V}^u_\Delta\| - (\upsilon_a + \upsilon_b)}$$

Proof. From proposition 1, we have:

$$t_{ca} = \frac{-(\mathbf{P}_\Delta.\mathbf{V}_\Delta)}{{\mathbf{V}_\Delta}^2} = \frac{|\mathbf{P}_\Delta.\mathbf{V}_\Delta|}{{\mathbf{V}_\Delta}^2}$$

Or

$$\frac{|\mathbf{P}_\Delta.\mathbf{V}_\Delta|}{{\mathbf{V}_\Delta}^2} \leq \frac{\|\mathbf{P}_\Delta\| \times \|\mathbf{V}_\Delta\|}{\|{\mathbf{V}_\Delta}^2\|} = \frac{\|\mathbf{P}_\Delta\|}{\|\mathbf{V}_\Delta\|}$$

and

$$\begin{aligned}\|\mathbf{P}_\Delta\| &= \|\mathbf{P}_a - \mathbf{P}_b\| \\ &= \|(\mathbf{P}_a^u - \mathbf{P}_b^u) + (\mathbf{P}_a - \mathbf{P}_a^u) + (\mathbf{P}_b^u - \mathbf{P}_b)\| \\ &\leq \|\mathbf{P}_\Delta^u\| + \rho_a + \rho_b\end{aligned}$$

$$\begin{aligned}\|\mathbf{V}_\Delta^u\| &= \|\mathbf{V}_a^u - \mathbf{V}_b^u\| \\ &= \|(\mathbf{V}_a - \mathbf{V}_b) + (\mathbf{V}_a^u - \mathbf{V}_a) + (\mathbf{V}_b - \mathbf{V}_b^u)\| \\ &\leq \|\mathbf{V}_\Delta\| + \upsilon_a + \upsilon_b\end{aligned}$$

The time of closest approach t_{ca} is consequently upper bounded by $\frac{\|\mathbf{P}_\Delta^u\|+(\rho_a+\rho_b)}{\|\mathbf{V}_\Delta^u\|-(\upsilon_a+\upsilon_b)}$ □

Proposition 6.

$$\textit{If } \exists\ t \in [0, T_l] /\ \|\boldsymbol{P}_\Delta + t\,\boldsymbol{V}_\Delta\| \leq R_a + R_b$$

$$\textit{Then } \exists\ t \in [0, T_l] /\ \|\boldsymbol{P}_\Delta^u + t\,\boldsymbol{V}_\Delta^u\| \leq R_a + R_b + R_{unc}$$

Where:

$R_{unc} = \rho_a + \rho_b + t^*(\upsilon_a + \upsilon_b)$

$$t^* = \begin{cases} min(T_l, \frac{\|\boldsymbol{P}_\Delta^u\|+(\rho_a+\rho_b)}{\|\boldsymbol{V}_\Delta^u\|-(\upsilon_a+\upsilon_b)}) & if \quad \|\boldsymbol{V}_\Delta^u\| > \upsilon_a + \upsilon_b \\ T_l & else \end{cases}$$

Proof.

$$\begin{aligned}\|\mathbf{P}_\Delta^u + t\mathbf{V}_\Delta^u\| &= \|\mathbf{P}_\Delta + t\mathbf{V}_\Delta + \mathbf{P}_\Delta^u - \mathbf{P}_\Delta + t(\mathbf{V}_\Delta^u - \mathbf{V}_\Delta)\| \\ &\leq \|\mathbf{P}_\Delta + t\mathbf{V}_\Delta\| + \|\mathbf{P}_\Delta^u - \mathbf{P}_\Delta\| + t\,\|\mathbf{V}_\Delta^u - \mathbf{V}_\Delta\|\end{aligned}$$

Or

$$\|\mathbf{P}_\Delta + t\mathbf{V}_\Delta\| \leq R_a + R_b \tag{4.12}$$

$$\|\mathbf{P}_\Delta^u - \mathbf{P}_\Delta\| = \|\mathbf{P}_a^u - \mathbf{P}_a + \mathbf{P}_b - \mathbf{P}_b^u\| \leq \rho_a + \rho_b \tag{4.13}$$

$$\|\mathbf{V}_\Delta^u - \mathbf{V}_\Delta\| = \|\mathbf{V}_a^u - \mathbf{V}_a + \mathbf{V}_b - \mathbf{V}_b^u\| \leq \upsilon_a + \upsilon_b \tag{4.14}$$

We also know that $0 \leq t \leq t_{ca} \leq T_l$. Using proposition 5 we hence obtain:

$$\begin{cases} 0 \leq t \leq min(T_l, \frac{\|\mathbf{P}^u_\Delta\|+(\rho_a+\rho_b)}{\|\mathbf{V}^u_\Delta\|-(\upsilon_a+\upsilon_b)}) & if \quad \|\mathbf{V}^u_\Delta\| > \upsilon_a + \upsilon_b \\ 0 \leq t \leq T_l & else \end{cases} \tag{4.15}$$

Proposition 6 follows from equations (4.12), (4.13), (4.14) and (4.15). □

Proposition 6 states that missed alarms caused by the state information uncertainty can be avoided by extending the alert threshold $R_a + R_b$ by the uncertainty distance R_{unc}. Indeed, R_{unc} represents an upper bound of the distance separating the certain relative position and the uncertain relative position at the time of closest approach t_{ca}. Unlike previous works that typically use static and experimentally predetermined uncertainty distances [Alej 09, Snap 10], our considered uncertainty distance R_{unc} is dynamically updated according to the current conflict geometry (i.e., current positions and velocities) and the uncertainties bounds. R_{unc} can consequently be adapted to any possible encounter geometry and measurements uncertainties.

4.5 Performance Evaluation

In this section we apply our analytical model to a swarm of small aerial vehicles moving in a 3D confined airspace. We consider three different sets of simulations: simulations in a perfect environment (i.e., with no packet loss and with exact state information), simulations with packet loss and finally simulations with uncertain state information.

4.5.1 Simulation setup

We have implemented SLIDE in the OMNeT++ simulator [Varg 01]. We consider N nodes moving for 10 minutes with a constant speed in a confined space represented by a mall of $500m$ x $400m$ x $30m$. With the constant speed mobility each node linearly moves with a constant speed towards a random target position. Once the target position is reached, a new random target is then calculated. We assume that, within a simulation run, all the nodes have the same velocity v, protected zone radius R_p and communication range R_c. Nodes can directly communicate with each other using the IEEE 802.11 protocol. Unless otherwise stated, the considered number of nodes, velocity, communication range and protected zone radius are set respectively to $N = 60$, $v = 3m/s$, $R_c = 25m$ and $R_p = 1m$. The maximal back-off time is set to $T_{off} = 1s$. It is worth mentioning that all the considered parameters in our simulations are consistent with the current specifications of small aerial vehicles. Each recorded data point is the average of at least

1000 independent simulation repetitions. This allows us to reach a 95% confidence interval and also simulate a wide range of possible conflicts (head-on conflicts, rear conflicts, 3-way conflicts, etc).

We adopt the following definitions:

- *Actual conflicts*: actual conflicts are the conflicts that actually occurred. They represent the set of conflicts that would be detected with a perfect conflict detection algorithm. The detection of the actual conflicts is crucial for our simulations since it enables us to assess the true positives, false negatives and false positives of our conflict detection algorithm. In our simulations, we used a central entity that is continuously tracking all the nodes mobility in order to detect the set of actual conflicts.

- *Predicted conflicts*: predicted conflicts are the conflicts predicted by the conflict detection algorithm.

- *Accurate alarms*: accurate alarms (or true positives) are predicted conflicts that have actually occurred.

Several metrics are used to evaluate the performance of the proposed algorithm:

- *Number of missed alarms*: missed alarms (or false negatives) are actual conflicts that the conflict detection algorithm failed to predict. The number of missed alarms is calculated as follows:

$$\#Missed\ alarms = \#Actual\ conflicts - \#Accurate\ alarms$$

- *Number of false alarms*: false alarms (or false positives; nuisance alarms) are predicted conflicts that have not actually occurred. The number of false alarms is calculated as follows:

$$\#False\ alarms = \#Predicted\ conflicts - \#Accurate\ alarms$$

- *Missed alarms probability*: missed alarms probability is the probability of a missed alarm given an actual conflict. It quantifies the likelihood of missing a conflict when it actually occurs. The missed alarms probability is estimated as follows:

$$Missed\ alarms\ probability = \frac{\#Missed\ alarms}{\#Actual\ conflicts}$$

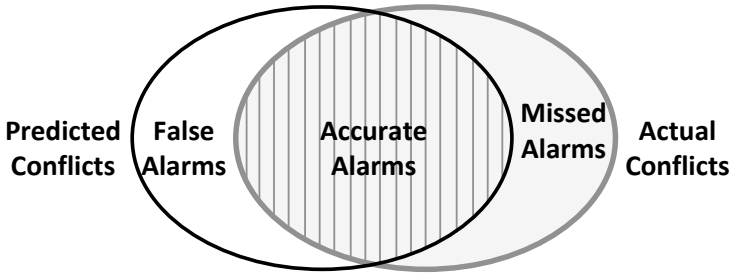


Figure 4.6 – Conflicts and alarms

- *False alarms probability*: false alarms probability is the probability of a false alarm given a predicted conflict. It quantifies the likelihood of predicting a conflict when it does not actually occur. The false alarms probability is estimated as follows:

$$\textit{False alarms probability} = \frac{\#False\ alarms}{\#Predicted\ conflicts}$$

- *Average manoeuvre time* t_m: the average manoeuvre time is the average time interval from the conflict alert issuing to the loss of separation. It quantifies the amount of lead-time the conflict detection algorithm provides for conflict resolution. The larger the manoeuvre time, the more time nodes have to avoid future conflicts.

Missed and false alarms measure the prediction accuracy of the conflict detection algorithm. To estimate them we must first determine the set of actual conflicts and then compare them to the set of predicted conflicts in order to finally determine the set of accurate alarms. A conceptual definition of missed and false alarms is represented in Figure 4.6. The role of our conflict detection algorithm is not limited to the determination of whether a pair of nodes are in a conflict course. It also provides timing information about the upcoming conflicts. This timing information must also be accurately determined. That is why, in contrast to major previous works that did not take the timing accuracy into consideration, we consider a predicted conflict as accurate only if the predicted start time of the conflict t_{pred} corresponds to the actual conflict start time t_{actual}. In this case, a given predicted conflict may be regarded as both a false and missed alarm. For example, if a conflict is predicted to start after it actually starts (i.e., $t_{pred} > t_{actual}$) then this prediction will be considered as a false alarm since the predicted time of conflict is not correct. The corresponding conflict will also be considered as a missed conflict since the conflict detection algorithm was not able to correctly predict it.

There was no attempt to resolve predicted conflicts during simulations because such an intervention corrupts the counting of false and missed alarms: there may be no actual conflicts since any predicted conflict would have been avoided prior to its occurrence.

Next, we consider the set of actual conflicts as our unique comparison reference. Indeed, the set of actual conflicts is the most representative comparison reference since it represents the set of conflicts that would be detected using a perfect detection technique. Besides, the comparison of our algorithm against previously proposed detection algorithms is not that significant as they adopted different assumptions (2D airspace, known trajectory plan, offline global path planning, different protected zones shapes, etc). Finally and most importantly, previously suggested techniques were restricted to the determination of whether a pair of airplanes are on a conflict course. They do not provide any information about the timing of the conflict. Timing information is nevertheless essential in order to match the alerts generated by the conflict detection algorithm to the actual conflicts: there may be multiple actual conflicts between the same pair of airplanes along their mission. Similarly, there may be multiple alerts issued by the conflict detection algorithm for the same pair airplanes.

4.5.2 Simulations in a perfect environment

In this subsection we first study the influence of the considered simulation parameters (i.e., the broadcast cycle T_b, the look ahead window T_l, the velocity v, the communication range R_c and the protected zone radius R_p) on the performance of our algorithm and then study its scalability in high density traffic scenarios.

Effect of the simulation parameters on SLIDE behaviour

In order to better understand the behaviour of SLIDE, we need first to study the effect of the different simulation parameters and particularly the tuning parameters T_b and T_l on SLIDE performance mainly measured by the number of missed alarms, false alarms and the manoeuvre times.

Figure 4.7 plots the number of missed alarms as a function of the broadcast cycle under two velocities $v = 3m/s$ and $v = 5m/s$. The protected zone radius is $R_p = 1m$ and the look ahead time is $T_l = 60s$. As expected, the number of missed alarms increases as the broadcast cycle increases. For example, for $v = 3m/s$, the number of missed alarms is of 6 alarms with a broadcast cycle of $6s$ and it increases to reach 12 missed alarms with a broadcast cycle of $9s$. It is also interesting to note that Figure 4.7 is consistent with equation (4.7). Indeed, from equation (4.7), to avoid missed alarms T_b should verify: $T_b \leq \frac{R_c - 2R_p}{2v} = \frac{23}{2v}$. For $v = 5m/s$, we should have $T_b \leq 2.3s$ and for $v = 3m/s$, we should have $T_b \leq 3.83s$. When referring to Figure 4.7, we see that for $v = 5m/s$ the number of missed alarms is negligible and barely increases for $T_b \leq 2.5s$ and then sharply and continuously increases for $T_b > 2.5s$. Similarly, for $v = 3m/s$, we observe an

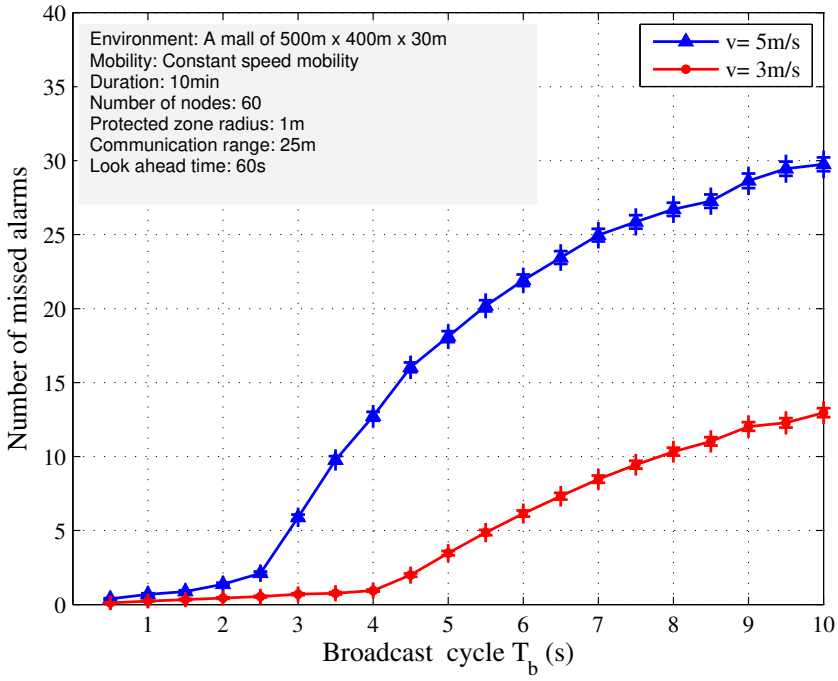


Figure 4.7 – Number of missed alarms under different broadcast cycles and velocities

identical behaviour for $T_b \leq 4s$ and $T_b > 4s$. It shall be noted, as shown in this figure, that unnecessarily small values of T_b have no positive effects on the number of missed alarms. It however unnecessarily increases the communication cost of our algorithm that corresponds to $\frac{N}{T_b}$ state messages per time unit, where N is the total number of deployed nodes.

Figure 4.8 plots the number of missed alarms as a function of the communication range under two velocities $v = 3m/s$ and $v = 5m/s$. The considered broadcast cycle is $T_b = 2s$ and the protected zone radius is $R_p = 1m$. From this figure, we clearly observe that the number of missed alarms decreases as the communication range increases. Indeed, the higher is the communication range, the farther gets the state information, and therefore the higher is the detection capability. Besides, equation (4.8) states that in order to prevent missed alarms, the communication range R_c should verify: $R_c \geq 2T_b v + 2R_p$, which corresponds to $R_c \geq 22m$ with a velocity $v = 5m/s$ and $R_c \geq 14m$ with a velocity $v = 3m/s$. In consistency with equation (4.8), for $v = 5m/s$, Figure 4.8 shows that the number of missed alarms continuously decreases as the communication range R_c increases from $5m$ to about $22m$ and then keeps almost stable and very low values. A similar behaviour is perceived with $v = 3m/s$ and $R_c = 14m$.

In Figure 4.9, we study the impact of the prediction time T_l, the communication range R_c and the protected zone radius R_p on the number of false alarms and the average

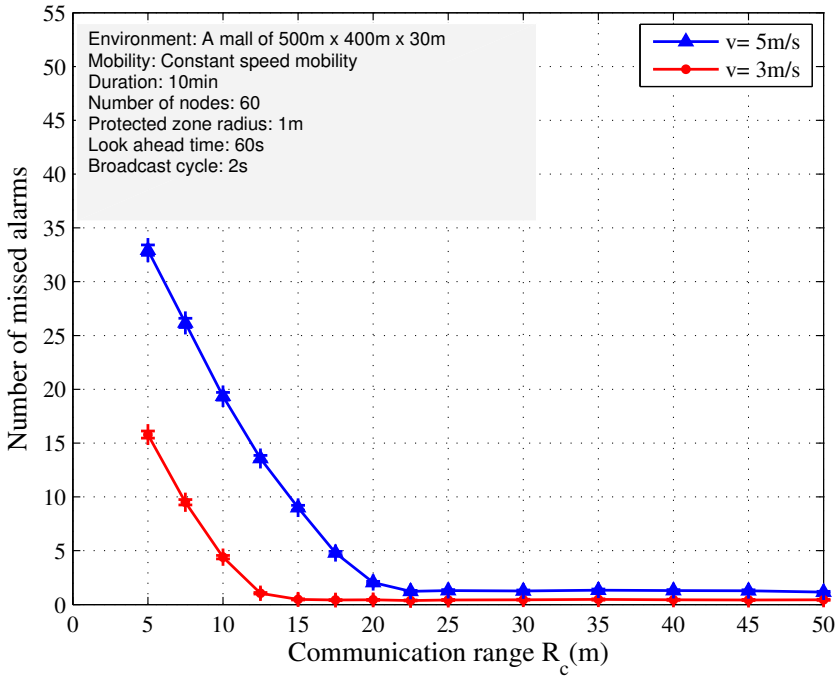


Figure 4.8 – Number of missed alarms under different communication ranges and velocities

manoeuvre time. The number of false alarms (Figure 4.9(a)) and the average manoeuvre time (Figure 4.9(b)) are plotted as a function of T_l, with R_c and R_p as parameters. We vary T_l from $2s$ to $60s$ and consider two communication ranges $R_c = 25m$ and $R_c = 65m$ and two protected zone radii $R_p = 1m$ and $R_p = 2.5m$.

As portrayed on Figure 4.9(a), the number of false alarms increases as T_l, R_c and R_p increase. For a given R_p and R_c, as the prediction time T_l increases, the number of false alarms increases and then converges to some maximum value for large prediction times. Take for example, a protected zone radius of $2.5m$ and a communication range of $25m$. Figure 4.9(a) shows that there were 2.2 false alarms with $T_l = 2s$, 8.5 false alarms with $T_l = 10s$ and 9.2 false alarms with $T_l = 20s$. The number of false alarms stabilizes at around 9.5 conflicts for $T_l \geq 40s$. This result is an indication that the performance of the conflict prediction directly depends on the accuracy of the trajectory prediction. Indeed, with the straight line trajectory prediction, the farther in advance a prediction is, the less certain such a prediction is likely to occur due to the higher probability of eventual unpredicted turning manoeuvres. Furthermore, for a fixed prediction time T_l and communication range R_c, the number of false alarms increases as the protected zone radius increases from $1m$ to $2.5m$. For instance, for $T_l = 10s$ and $R_c = 25m$, there were 1.5 false alarms with $R_p = 1m$ and 8.5 false alarms with $R_p = 2.5m$. This increase can be explained by the fact that an increased protected zone radius highly increases the number

of actual conflicts (we have observed 27 actual conflicts with a protected zone radius of $1m$ and 164 conflicts with a protected zone radius of $2.5m$). An increased number of actual conflicts increases the number of predicted conflicts which in its turn increases the likelihood of false predictions. Similarly, for a fixed prediction time T_l and protected zone radius R_p, the number of false alarms increases as the communication range increases from $25m$ to $65m$. When the communication range is increased from $25m$ to $65m$, each node will have more neighbouring nodes to predict conflicts with, which increases the likelihood of false predictions.

From Figure 4.9(b), we observe that the average manoeuvre time t_m is closely related to T_l, R_c and R_p: t_m increases as T_l and R_c increase and decreases as R_p increases. In fact, t_m is limited by a maximum value determined by T_l, R_c, R_p and the average relative speed v^* as follows:

$$t_m \leq min(T_l, \frac{R_c - 2R_p}{v^*})$$

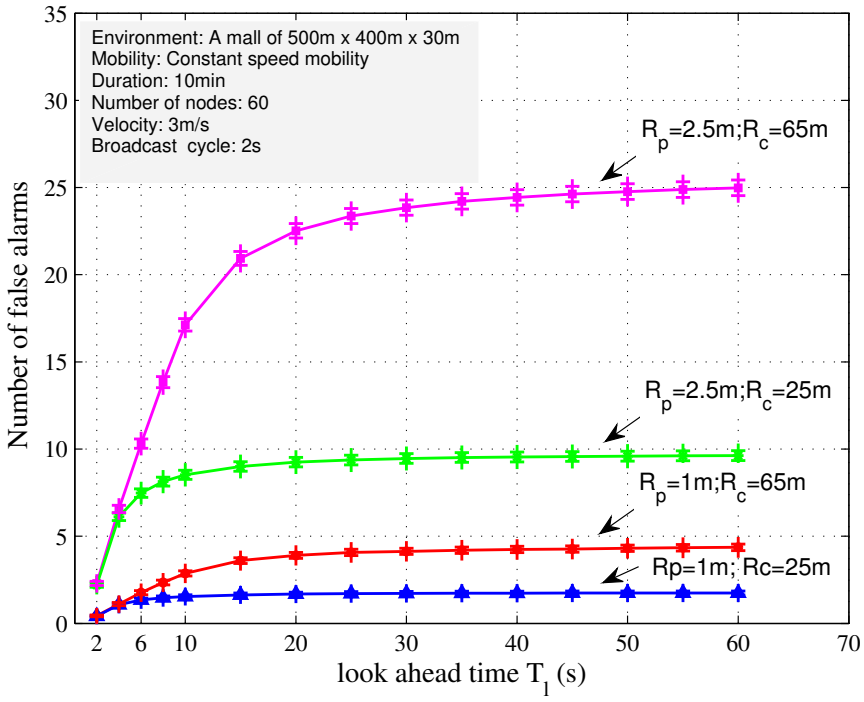
The expression $\frac{R_c-2R_p}{v^*}$ is the maximum possible time elapsed between the reception of a STATE message and the loss of separation. The relative speed v^* can be expressed as a direct function of the velocity v: $v^* = k \times v$, where k is a constant specific to the mobility model. For our considered mobility model, the parameter k is equal to $\frac{4}{\pi}$ [Groe 05]. We consequently have:

$$t_m \leq min(T_l, \frac{(R_c - 2R_p)\ \pi}{4\ v})$$

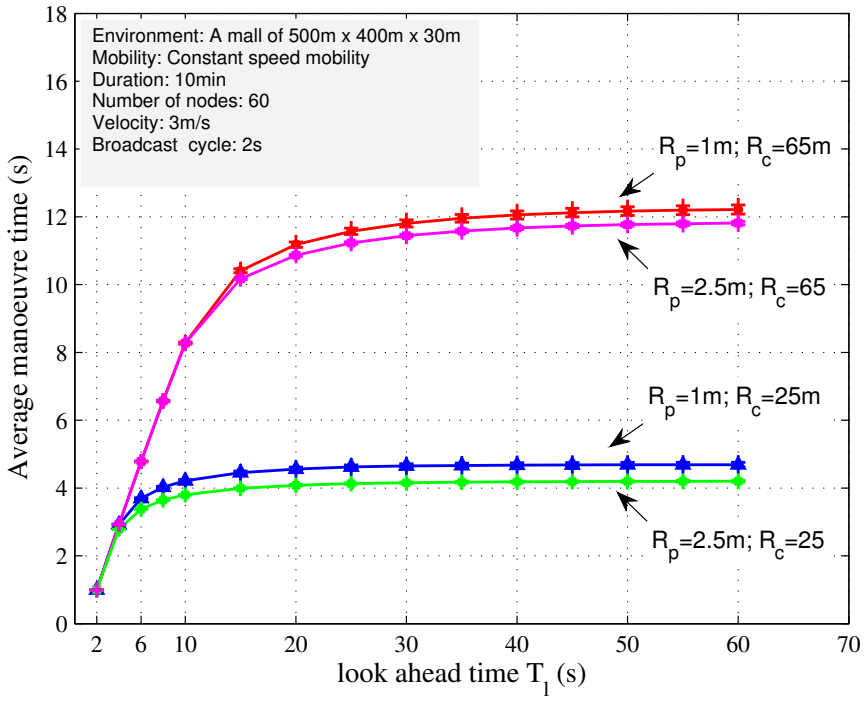
This corresponds to $t_m \leq min(T_l, 6.02s)$ with $R_c = 25m$ and $R_p = 1m$; $t_m \leq min(T_l, 5.23s)$ with $R_c = 25m$ and $R_p = 2.5m$; $t_m \leq min(T_l, 16.49s)$ with $R_c = 65m$ and $R_p = 1m$ and $t_m \leq min(T_l, 15.70s)$ with $R_c = 65m$ and $R_p = 2.5m$. These results are in conformity with the results obtained in Figure 4.9(b).

It can be concluded from the analysis of Figure 4.9 that the choice of T_l and R_c is a trade-off between nuisance alarms and large safety margins. Increasing T_l or R_c increases the manoeuvre times, but with the penalty of an increased number of false alarms. Decreasing T_l or R_c lowers the number of false alarms but will also lessen the manoeuvre times that may become insufficient for effective avoidance manoeuvres.

Table 4.2 shows the number of false alarms obtained under different confined airspaces dimensions X, Y, Z. We consider two look ahead times $T_l = 2s$ and $T_l = 60s$ and two velocities $v = 3m/s$ and $v = 5m/s$. The considered broadcast cycle is $T_b = 2s$. We can observe from this table that the number of false alarms decreases as the airspace size increases and the velocity decreases. This is because, in a large airspace the mobile nodes can have long straight line trajectories while in a small airspace with high velocities, the nodes will rapidly reach the environment boundaries and will hence have to frequently



(a)



(b)

Figure 4.9 – Number of false alarms (a) and average manoeuvre time (b) under different look ahead times, communication ranges and protected zone radii

$X \times Y \times Z$	$T_l = 2s$ & $v = 3m/s$	$T_l = 2s$ & $v = 5m/s$	$T_l = 60s$ & $v = 3m/s$	$T_l = 60s$ & $v = 5m/s$
$400 \times 300 \times 20$	1.35	2.98	5.96	7.65
$400 \times 300 \times 25$	1.05	2.65	4.53	6.61
$450 \times 350 \times 25$	1.04	1.7	3.16	4.36
$450 \times 350 \times 30$	0.59	1.63	2.27	3.92
$500 \times 400 \times 30$	0.52	1.07	1.81	2.45
$500 \times 400 \times 35$	0.48	0.88	2.03	2.1
$550 \times 450 \times 35$	0.35	0.61	1.26	1.64

Table 4.2 – Number of false alarms under different spaces

change their movement direction. For example, there were 7.65 false alarms in an airspace size of $400m$ x 300 x $20m$ with a velocity $v = 5m/s$ and a look ahead time $T_l = 60s$. This number decreases to 5.96 with a velocity $v = 3m/s$ and to 1.64 false alarms in an airspace of $550m$ x 450 x $35m$. Short look ahead times should be used in small deployment areas in order to avoid an increasing number of false alarms. As shown in the table, a short look ahead time $t_l = 2s$ guarantees a low number of false alarms even in small deployment areas with high velocities.

SLIDE scalability in a homogeneous network

Table 4.3 studies the scalability of SLIDE in a homogeneous network where all the deployed nodes have the same characteristics. It shows the number of actual conflicts, the number of missed alarms, the number of false alarms, the probability of missed alarms, the probability of false alarms and the average manoeuvre time under a number N of nodes varying from 30 to 120. The protected zone radius R_p is equal to 1m. We consider three velocities $v = 1m/s$, $v = 3m/s$ and $v = 5m/s$. We have observed from the previous simulations that the performance of SLIDE is mainly influenced by its tuning parameters T_b and T_l. These two parameters should be carefully tuned in order to guarantee an efficient conflict detection. Based on equation (4.7) and the previous simulation results the values of the tuning parameters are set to: $T_b = 12s$ for $v = 1m/s$, $T_b = 3.5s$ for $v = 3m/s$, $T_b = 2s$ for $v = 5m/s$ and $T_l = 60s$ for the three considered velocities.

Table 4.3 shows an important increase in the number of actual conflicts when the number of nodes increases from 30 to 120: the number of actual conflicts increased from 3.49 to 35.77 under the velocity $v = 1m/s$, from 7.48 to 109.02 under the velocity $v = 3m/s$ and from 11.31 to 180.36 under the velocity $v = 5m/s$. In spite of this important increase in the number of actual conflicts, the probability of missed alarms and the probability of false alarms remained almost constant. SLIDE maintained a constant and a very low missed alarms probability of around 0.03 and a very low and constant false alarms probability of around 0.06 under the three considered velocities.

Predictions should not only be accurate regarding the existence of conflicts. Conflicts

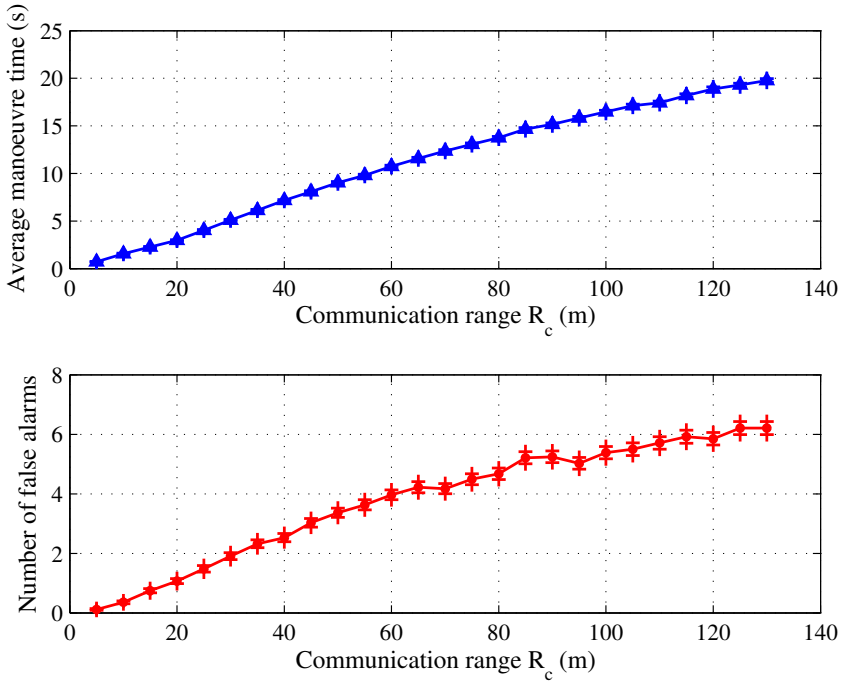


Figure 4.10 – Average manoeuvre time and number of false alarms as a function of the communication range

should also be predicted sufficiently in advance so that avoidance manoeuvres get enough time to be performed. Table 4.3 shows that SLIDE maintained a manoeuvre time of around $12s$ under the velocity $v = 1m/s$, $4s$ under the velocity $v = 3m/s$ and $2.4s$ under the velocity $v = 5m/s$, which is suitable in regard to the considered velocities and protected zone radius. Indeed, the higher is the velocity, the more rapidly nodes can manoeuvre to escape the collision risk and thus the lower is the required manoeuvre time. However, if the end user desires larger manoeuvre times, he/she can increase the communication range but at the expense of a slightly higher number false alarms. This can be seen in Figure 4.10 that plots the average manoeuvre time and the number of false alarms under a varying communication range. For instance, Figure 4.10 shows that to increase the average manoeuvre time from 4.02 to $10s$, the communication range should be increased from $25m$ to $55m$ but on the other hand the number of false alarms will increase from 1.48 to 3.63.

Overall, table 4.3 clearly shows that SLIDE is scalable to an increasing number of nodes which is an advantageous feature of a conflict detection algorithm.

$V = 1m/s$							
N	30	45	60	75	90	105	120
Actual conflicts	3.49	5.5	9.20	14.43	20.20	27.81	35.77
Number of missed alarms	0.09	0.10	0.16	0.33	0.51	0.89	1
Number of false alarms	0.06	0.25	0.45	0.59	0.93	1.24	1.94
Probability of missed alarms	0.026	0.019	0.018	0.023	0.025	0.032	0.027
Probability of false alarms	0.017	0.044	0.047	0.040	0.045	0.044	0.052
Average manoeuvre time t_m(s)	12.60	11.97	11.78	11.79	11.59	11.44	11.60
$V = 3m/s$							
N	30	45	60	75	90	105	120
Actual conflicts	7.48	15.54	27.10	41.87	60.12	84.3	109.02
Number of missed alarms	0.22	0.39	0.76	1.19	1.71	2.36	3.16
Number of false alarms	0.35	0.80	1.48	2.37	3.59	5.24	6.1
Probability of missed alarms	0.030	0.025	0.028	0.028	0.028	0.028	0.028
Probability of false alarms	0.046	0.050	0.053	0.055	0.057	0.06	0.054
Average manoeuvre time t_m(s)	3.98	4.07	4.02	4.03	4.14	3.98	4.00
$V = 5m/s$							
N	30	45	60	75	90	105	120
Actual conflicts	11.31	24.58	44.46	70.30	101.67	137.92	180.36
Number of missed alarms	0.30	0.71	1.38	2.12	2.97	3.87	5.13
Number of false alarms	0.56	1.46	2.45	4.10	6.05	8.62	10.75
Probability of missed alarms	0.026	0.028	0.031	0.030	0.029	0.028	0.028
Probability of false alarms	0.048	0.057	0.053	0.056	0.057	0.060	0.057
Average manoeuvre time t_m(s)	2.50	2.39	2.49	2.41	2.49	2.43	2.43

Table 4.3 – SLIDE scalability in a homogeneous network: SLIDE guarantees a very low probability of missed alarms, a very low probability of false alarms and an adequate manoeuvre time even in high density traffic scenarios

SLIDE scalability in a heterogeneous network

Some applications require the deployment of different nodes with different characteristics and particularly different velocities and sizes. We therefore study next the performance of SLIDE in a heterogeneous network where nodes have different velocities and protected zones radii (i.e., sizes). We assume that the deployed nodes are divided into three different groups ($G1$, $G2$,$G3$) and each group has its own characteristics. We consider the following deployment scenarios:

- Scenario 1: group 1 ($v = 1m/s$, $R_p = 1$), group 2 ($v = 3m/s$, $R_p = 1$), group 3 ($v = 5m/s$, $R_p = 1$)
- Scenario 2: group 1 ($v = 3m/s$, $R_p = 0.5$), group 2 ($v = 3m/s$, $R_p = 1$), group 3 ($v = 3m/s$, $R_p = 1.5$)
- Scenario 3: group 1 ($v = 1m/s$, $R_p = 0.5$), group 2 ($v = 3m/s$, $R_p = 1$), group 3 ($v = 5m/s$, $R_p = 1.5$)

The extension of equation (4.6) to a heterogeneous network gives:

$$T_b \leq \frac{R_{c_{min}} - 2R_{p_{max}}}{2V_{max}} \tag{4.16}$$

Where $R_{c_{min}}$ is the minimum communication range in the network, $R_{p_{max}}$ the maximum protected zone radius and V_{max} the maximum velocity.

According to equation (4.16), we choose $T_b = 2s$ for the first and the third scenario and $T_b = 3.5s$ for the second scenarios. The look ahead time is fixed to 60s and the number of nodes is varied from 30 to 120 for the three of the considered scenarios.

Table 4.4 summarizes the obtained results. As shown in this table, SLIDE maintains constant and very low missed and false alarms probabilities although the increase of the number of nodes. SLIDE also maintains constant manoeuvre times of around $4s$ under the three considered scenarios. Once again, our algorithm shows its scalability to an increasing number of nodes even in heterogeneous networks.

Manoeuvre times and conflict resolution

The main focus of this chapter is on conflict detection. We can however, with a closer focus on the obtained manoeuvre times, provide some general indications on possible resolution manoeuvres. Indeed, when we refer to table 4.5 that summarizes the obtained manoeuvre times under different velocities, we can see that SLIDE provides an average manoeuvre distance around $12m$ under all the considered velocities. This corresponds to around 12 times the considered protected zone radius. With such a large manoeuvre

Scenario 1							
N	30	45	60	75	90	105	120
Actual conflicts	8.08	17.48	30.55	47.23	68.96	93.17	122.63
Number of missed alarms	0.11	0.28	0.43	0.68	1.17	1.52	1.95
Number of false alarms	0.49	0.89	1.59	2.62	4.08	5.42	7.25
Probability of missed alarms	0.014	0.016	0.014	0.014	0.017	0.016	0.015
Probability of false alarms	0.058	0.049	0.050	0.053	0.056	0.055	0.056
Average manoeuvre time t_m(s)	4.17	4.07	4.19	4.31	4.22	4.21	4.29
Scenario 2							
N	30	45	60	75	90	105	120
Actual conflicts	7.66	16.38	19.65	45.63	65.42	91.32	117.42
Number of missed alarms	0.24	0.38	0.50	1.29	2.05	2.52	3.53
Number of false alarms	0.39	0.83	0.96	2.52	3.63	5.32	6.73
Probability of missed alarms	0.031	0.023	0.025	0.028	0.031	0.027	0.030
Probability of false alarms	0.050	0.049	0.047	0.053	0.054	0.057	0.055
Average manoeuvre time t_m(s)	3.93	3.99	4.07	4.00	4.08	4.00	3.93
Scenario 3							
N	30	45	60	75	90	105	120
Actual conflicts	10	22.19	38.16	58.89	87.22	117.05	155.41
Number of missed alarms	0.11	0.41	0.80	1.01	1.83	2.24	2.84
Number of false alarms	0.58	1.06	2.07	3.1	5.08	6.44	8.86
Probability of missed alarms	0.011	0.018	0.020	0.017	0.021	0.019	0.018
Probability of false alarms	0.056	0.046	0.052	0.050	0.056	0.053	0.054
Average manoeuvre time t_m(s)	3.53	3.38	3.45	3.40	3.45	3.47	3.40

Table 4.4 – SLIDE scalability in a heterogeneous network: SLIDE maintains very low missed and false alarms probabilities and adequate manoeuvre times even in highly dense heterogeneous traffic scenarios

	$v = 1m/s$	$v = 3m/s$	$v = 5m/s$
Manoeuvre time t_m (s)	11.98	4.02	2.49
Manoeuvre distance $= t_m \times v$	11.98	12.06	12.45

Table 4.5 – Manoeuvre times under different velocities

distance, a conflict can be easily resolved using a single basic horizontal manoeuvre (a turn left or right with a distance $d > R_p$) or a single basic vertical manoeuvre (a climb or descent with a distance $d > R_p$). For instance, to resolve a head on conflict the higher node can realize a climb manoeuvre while the lower node continues its trajectory without change or vice versa. To resolve a crossing conflict the node on the left can realize a turn left manoeuvre while the node on the right continues its trajectory without change or vice versa.

To summarize, the simulation results obtained in subsection 4.5.2 clearly indicate that:

1. The performance of SLIDE is mainly impacted by its two tuning parameters T_b and T_l.

2. An adequate tuning of these two parameters guarantees excellent conflict detection capabilities. Indeed, as observed from tables 4.3 and 4.4, even in high density traffic scenarios SLIDE provides:

   - A very low probability of missed alarms around 0.03.
   - A very low probability of false alarms around 0.06.
   - An adequate manoeuvre time with respect to the nodes velocities and protected zones radii.

4.5.3 Simulations with packet loss

In Figure 4.11, we study the impact of packet loss on the number of missed alarms. The number of missed alarms is plotted as a function of the packet loss probability p under different broadcast cycles T_b. The packet loss probability p is varied from 0 to 0.5. As expected, the number of missed alarms increases as the packet loss probability increases. Figure 4.11 also shows that the higher the broadcast cycle, the higher the negative effect of packet loss on the detection performance. For example with a packet loss probability $p = 0.4$ (a highly perturbed environment), there were 0.2 missed alarms with $T_b = 0.5s$, 1.2 missed alarms with $T_b = 1.5s$, 5.4 missed alarms with $T_b = 2.5s$ and 8.4 missed alarms with $T_b = 3.5s$.

To validate proposition 4, we present in table 4.6 the number of missed alarms under different distinguished broadcast cycles T_b and $T_b^*[u]$. $T_b = \frac{R_c - 2R_p}{2v} = 3.8s$ corresponds to

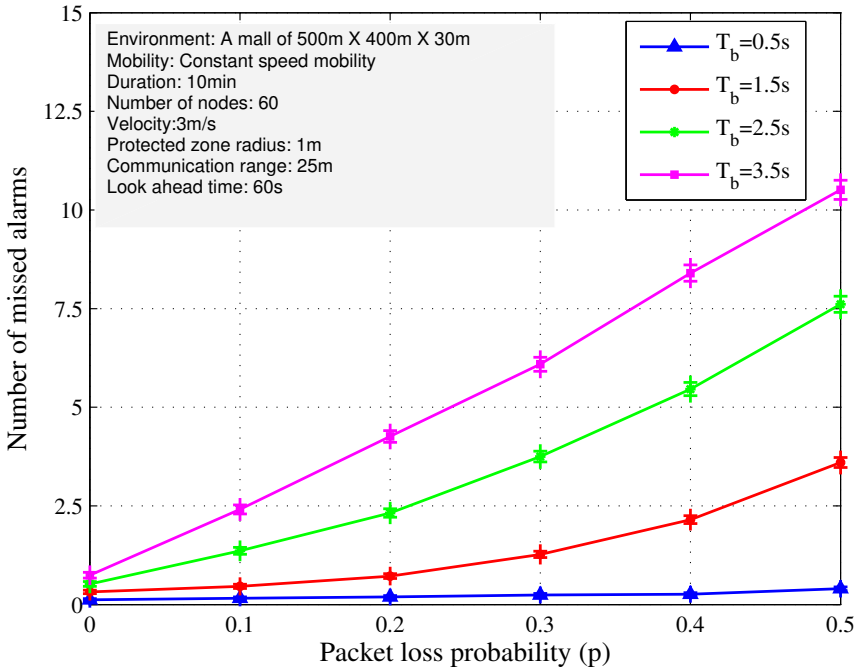


Figure 4.11 – Number of missed alarms under different packet loss probabilities and broadcast cycles

the endpoint of equation (4.7) and $T_b^*[u] = \frac{\log(p)}{\log(1-u)} \times \frac{R_c - 2R_p}{2v} = \frac{\log(p)}{\log(1-u)} \times 3.8s$ corresponds to the endpoint of equation (4.11) in proposition 4. We consider three different probabilities: $u = 0.99$, $u = 0.95$ and $u = 0.90$. The packet loss probability p is varied from 0.1 to 0.5.

In our simulations, we restricted our attention to the case where the nodes have the same fixed packet loss probability during their entire flight. The packet loss probability in proposition 4 may nevertheless be a fixed or a dynamic packet loss probability. In case of dynamic packet loss probabilities, the nodes can dynamically, within the same flight, adapt their broadcast cycles to the loss probabilities.

As seen from table 4.6, using a loss-probability dependent broadcast cycles, T_b^*, with high u guarantees a negligible number of missed alarms in spite of the increasing packet loss probability p. However, with a fixed broadcast cycle independent from the probability of packet loss, T_b, we have a high increasing number of missed alarms.

To sum up, the simulation results obtained in subsection 4.5.3 indicate that in a packet-loss prone environment:

1. The use of an inadequately fixed broadcast cycle T_b leads to an increase in the number of missed alarms.

2. The use of a loss-probability dependent broadcast cycle T_b^* calculated according to proposition 4 guarantees a low number of missed alarms even in a highly perturbed

	$T_b = 3.8s$	$T_b^*[0.99]$	$T_b^*[0.95]$	$T_b^*[0.90]$
$p = 0.1$	2.62	0.55	1.76	2.51
$p = 0.2$	4.56	0.52	1.44	3.1
$p = 0.3$	6.57	0.42	1.43	2
$p = 0.4$	8.98	0.59	1.23	1.91
$p = 0.5$	11.54	0.51	1.13	2.2

Table 4.6 – SLIDE performance in a packet-loss prone environment: SLIDE guarantees a very low number of missed alarms when using a loss-probability dependent broadcast cycle with $u = 0.99$

communication environment. We particularly recommend the use of proposition 4 with a probability $u = 0.99$. Indeed, as we have seen in table 4.6, with $u = 0.99$ SLIDE provides a very low number of missed alarms around 0.5.

4.5.4 Simulations with state information errors

In previous simulations, we assumed that each node is aware of its exact state information. Here, we relax this assumption and introduce measurement errors in each node state information. The considered position and velocity errors are respectively randomly and uniformly distributed in $[0, \rho]$ and $[0, \upsilon]$. That is, for each node, we have: $\|\mathbf{P} - \mathbf{P}^u\| \leq \rho$ and $\|\mathbf{V} - \mathbf{V}^u\| \leq \upsilon$, were $\mathbf{P}$ and $\mathbf{P}^u$ are the node certain and uncertain position vectors; $\mathbf{V}$ and $\mathbf{V}^u$ are the node certain and uncertain velocity vectors. We differentiate between two different separation thresholds: the *minimum required separation* D_{min} and the *alert threshold* S. The *minimum required separation* D_{min} guarantees the safety of the mobile nodes and should never be violated. An actual conflict occurs when the distance separating two nodes is less or equal to $D_{min} = 2R_p$. The *alert threshold* S is the threshold considered to issue conflict alerts. The conflict detection algorithm declares a conflict if $\exists\ t \in [0, T_l]$ such that the distance separating two nodes is less or equal to S.

Table 4.7 summarizes the obtained simulation results under different position and velocity errors. The first three upper columns of the table correspond to the obtained number of missed alarms, false alarms and manoeuvre times in the case of using the intuitive alert threshold $S = D_{min} = 2R_p$. In this case, we observe that both the number of missed alarms and false alarms increase as the state information uncertainty increases. The average manoeuvre times, however, have slightly decreased in comparison with those obtained in an error-less environment (Table 4.3). SLIDE detection capability cannot consequently be maintained using such a basic threshold. Missed alarms are considered as extreme hazards that may lead to nodes collisions while false alarms are considered as nuisance alarms. Achieving low numbers of missed alarms is consequently a priority in

order to guarantee the nodes safety. An acceptable number of false alarms is also required in order not to loose the operator trust.

The next three upper columns of the table present the performance of SLIDE when we consider the dynamically extended alert threshold $S = 2R_p + R_{unc}$ introduced in proposition 6. As shown in the table, this method keeps a fairly constant and negligible number of missed alarms despite the increasing measurement errors. It also maintains quite constant manoeuvre times of about 4s. Note also that the obtained number of missed alarms and manoeuvre times are similar to those obtained in an ideal environment with no state information uncertainties (Table 4.3). This extended alert threshold effectively decreases the number of missed alarms but also remarkably increases the number of false alarms. Indeed, a non violation of the extended alert threshold $S = 2R_p + R_{unc}$ necessarily means a non violation of the minimum required separation $D_{min} = 2R_p$, but on the other hand a violation of S does not necessarily mean a violation of D_{min}.

To reduce this excessive number of false alerts, we propose the use of a two-stage validation. That is, for a conflict alert to be issued, each conflict must be validated upon two successive STATE messages. As we can see from the table, this method reduces the excessive number of false alarms while preserving a negligible number of missed alarms. Another observation is that the average manoeuvre times have slightly decreased in comparison with the previous method. This decrease is due the fact that the two-stage validation defers each conflict alert till the next received STATE message. The number of false alarms achieved under the two-stage method can be further reduced by using shorter look ahead times. The last three lower columns of the table present the obtained results with a shortened look ahead time $T_l = 5s$. The number of false alarms has significantly decreased but at the cost of shorter manoeuvre times.

Overall, the simulation results obtained in subsection 4.5.4 indicate that:

1. SLIDE performance degrades when it uses the basic alert threshold $S = 2R_p$ under state information uncertainties.

2. The algorithm performs perfectly well with respect to missed alarms and manoeuvre times when it uses the dynamically extended alert threshold $S = 2R_p + R_{unc}$. Nevertheless, extending the alerting threshold leads to an important increase in the number of false alarms. This increased number of false alarms can be significantly reduced using a two-stage validation and/or a short look ahead time but with the penalty of shorter, yet still adequate, manoeuvre times. In order for SLIDE to maintain good detection capabilities in an environment with sensing uncertainties it is therefore highly recommended to use both the dynamically extended alert threshold introduced in proposition 6 and the two-stage validation.

	$S = 2R_p$			$S = 2R_p + R_{unc}$		
	Missed	False	t_m	Missed	False	t_m
$\rho = 0.1$; $\upsilon = 0$	1.79	3.18	3.77	0.78	6.83	3.98
$\rho = 0.3$; $\upsilon = 0$	5.02	7.61	3.65	0.71	15.05	4.13
$\rho = 0.1$; $\upsilon = 0.1$	5.12	10.76	3.22	0.64	44.49	4.08
$\rho = 0.3$; $\upsilon = 0.1$	6.85	13.12	3.27	0.73	56.12	4.01
	$S = 2R_p + R_{unc}$ & two-stage validation			$S = 2R_p + R_{unc}$ & two-stage validation & shortened T_l		
	Missed	False	t_m	Missed	False	t_m
$\rho = 0.1$; $\upsilon = 0$	0.90	5.64	3.36	0.96	4.57	2.22
$\rho = 0.3$; $\upsilon = 0$	0.92	11.23	3.35	0.93	8.88	2.30
$\rho = 0.1$; $\upsilon = 0.1$	0.87	26.17	3.32	0.89	15.53	2.36
$\rho = 0.3$; $\upsilon = 0.1$	0.78	33.43	3.30	0.93	20.15	2.44

Table 4.7 – SLIDE under state information uncertainties: SLIDE good performance is guaranteed by the use of the dynamically extended alert threshold and the two-stage validation

4.6 Conclusion

In this chapter, we have presented SLIDE a novel 3D, distributed and straight line conflict detection and alerting algorithm. The proposed algorithm requires little communication between the nodes: only the state information (i.e., 3D position and velocity vectors) is periodically exchanged. Conflict conditions were first discussed for a perfect environment and then discussed in the presence of packet loss and state information uncertainties. State information uncertainties were handled by using a dynamically extended alert threshold and a two-stage conflict validation. Packet loss was handled using a loss-probability dependent broadcast cycle. To evaluate the performance of SLIDE, we carried out three sets of simulations: in an error-free environment, with packet loss and finally with state information uncertainties. Simulations results showed that SLIDE performs well in each of these cases.

Chapter 5

Probabilistic Conflict Detection in Mobile Wireless Networks

Contents

In this chapter, we introduce a stochastic model that accurately models conflicts in a swarm of 3D-mobile nodes sharing the same airspace. Unlike the most existing stochastic models, our proposed model is simple, generic and depends only on two input parameters: the number of nodes N and the parameter λ describing the time that two random nodes take to come into contact with each other.

The major contributions of this chapter can be summarized as follows:

- Based only on two parameters (N and λ), we provide simple yet accurate closed-form expressions for various conflict related metrics.
- The parameter λ, characterizing the inter-contact time between a pair of mobile nodes, may be difficult to obtain experimentally. We hence provide a generic explicit expression of λ. The obtained generic expression is then specified for two widely used mobility models: the random waypoint and the random direction mobility models.
- We validate the obtained analytical expressions based on extensive simulations using OMNeT++.

This chapter is organized as follows. Section 5.1 critically reviews state-of-the-art probabilistic conflict detection techniques. Section 5.2 presents our analytical model. We particularly consider two scenarios. In the first scenario, we assume that mobile nodes have perfect conflict detection and avoidance capabilities. In the second scenario, we assume that the mobile nodes are deprived of conflict detection and avoidance capabilities due to size, weight or power constraints. Section 5.3 validates the analytical results by comparing them against simulation results obtained under the random waypoint and the random direction mobility models. The simulation results are in excellent agreement with the analytical result. Besides, an explicit expression is provided for the parameter λ for the two considered mobility models. Section 5.4 concludes the chapter.

5.1 Probabilistic Conflict Detection Techniques

Conflict risk assessment techniques could be temporal, spatial or probabilistic. Temporal and spatial approaches compute the time and space coordinates of potential conflicts. SLIDE our conflict detection algorithm proposed in chapter 4 falls in this category. Probabilistic approaches [Liu 11, Hard 13] use analytic, numerical approximation and simulation techniques to diagnose conflict risks. [Leve 11] develops a generic mathematical model predicting the rate of conflicts that would occur in a 2D swarm of vehicles when no conflict avoidance mechanism is implemented. The conflict rate is expressed as a function of many parameters that must be defined by the swarm designer and identified based on vehicles behaviors knowledge. The proposed model is hence unpractical in case of non-availability of a priori estimates of these parameters. The works in [Blom 06] and [Stro 09] use Monte Carlo methods for the assessment of conflict risks. While Monte Carlo approaches represent a powerful tool to deal with nonlinear systems, their computation times may be very increased which makes them inefficient for real-time conflict warnings. Monte Carlo approaches also do not guarantee a non-underestimation of the conflict risk

resulting in violations of the safety requirements. [Pati 12] and [Lamb 08] propose formulations to compute the probability of conflicts between a moving vehicle and a set of static obstacles. In [Pati 12] the vehicle is assumed to be operating under Gaussian motion with uncertainties. The estimates of the conflict probability are based on a priori distributions of the vehicle state along a given plan. The proposed method ensures conservatives estimates: the probability of conflicts is never underestimated in order to guarantee the safety requirements. However, this approach assumes a precise sensing of all the obstacles in the environment which may not hold in the real world applications. In [Lamb 08] the probability of conflicts is expressed as a sum of integral of a product of Gaussian. The obtained integrals consider the state uncertainties of both the vehicle and the obstacles. The major drawback of this work is that the proposed formulation is complicated and has no analytical solution. The authors have instead proposed a Monte Carlo method to compute the integrals. This formulation was later extended in [Du T 11] to take into account the dependence between the vehicles and obstacles distributions (e.g., interactive vehicle-obstacle scenarios). Paper [Shyu 08] analyzes real-world data records in order to assess the conflicts risks. Real-world data encloses the major factors affecting the vehicles trajectories and hence produces reliable estimations but the data collection and analysis is usually memory and time consuming. The authors of [Saha 15] propose an approach that estimates the probability of conflict risk between a pair of vehicles moving in a 2D plane. The proposed method has short runtimes which makes it attractive for real-time conflict risks assessment but has also a limited applicability to 2D movements.

The previously proposed probabilistic conflict assessment techniques are usually complicated, time consuming and require many input parameters. Taking into consideration these issues, we propose in this chapter a simple stochastic model that accurately models conflicts risks using only two input parameters.

5.2 Analytical Model

In this section we begin with some definitions and a description of the considered network. We then consider two possible scenarios, a scenario where the mobile nodes are supposed to be equipped with perfect detection and avoidance capabilities and a scenario where the mobile nodes are not equipped with detection and avoidance capabilities due to their restricted size, weight or power.

5.2.1 Network description and definitions

We consider here an ad-hoc aerial wireless network composed of N nodes enumerated as $u_1, u_2, \cdots, u_N$. The scenario is that of an opportunistic network with a relatively sparse node density and intermittent connectivity. We assume that two mobile nodes are in *contact* if their distance is less than or equal to a parameter r_c, called the *contact range*. Whenever a given node u_i penetrates inside the sphere centered at node u_j with radius r_c, we say that a contact has occurred between u_i and u_j yielding to a *conflict*. A conflict may result in potential nodes damages, unless otherwise avoided at the cost of additional conflict detection and avoidance algorithms [Lam 09], which is costly in terms of complexity and energy.

In order to address the conflict risks between two mobile nodes, we borrow the following definitions from opportunistic networks. The *contact time* is defined as the duration between the instants two mobile nodes move into contact range r_c until they move apart. The *inter-contact time* between two mobile nodes is defined as the elapsed time between two consecutive contacts. Note that in conventional opportunistic networks, the above time-based metrics are rather defined with respect to the communication range ρ, since the interest in such networks stems from the fact that mobile nodes collaborate in distributing content when two nodes are within each others communication range. While a contact between mobile nodes in opportunistic networks is a *desirable* event for content distribution, it becomes a threatening event in aerial wireless networks since it indicates a potential collision risk between two mobile nodes. Therefore, contacts occurrence with respect to the contact range r_c should be avoided in such networks.

5.2.2 Inter-contact time characterization

There has been a lot of research work characterizing the inter-contact time in mobile opportunistic networks. This characterization is essential for the performance evaluation of the systems. Previous works have studied the distribution of the inter-contact time by collecting data from real mobile network environments. Some work has shown that the inter-contact time distribution is exponential with rate λ for both human [Kara 10] and some vehicles mobility scenarios [Zhu 10]. Nevertheless, there is some controversy about whether this exponential distribution relates to all real mobility patterns [Hui 05]. Some empirical results have shown that the aggregate inter-contact time distributions follows a power-law distribution and has a long tail [Chai 07], meaning that some pairs of nodes barely experience any contact.

In our model, we extend the exponential inter-contact distribution assumption to ad-hoc aerial wireless networks. Hence, we assume that the mobility of nodes is such that the

inter-contact times between the same pair of nodes –also called the pairwise inter-contact time– can be modeled by independent identically distributed (*i.i.d*) random variables that are exponentially distributed. We also assume that nodes in the network are homogeneous, that is, all the nodes have the same movement patterns that follows the same exponential inter-contact distribution with average contact rate λ. Let T be the pairwise inter-contact time between two given nodes. Therefore, the Cumulative Distribution Function (*CDF*) of random variable T is given by

$$\Pr(T \leq t) = 1 - e^{-\lambda t}. \tag{5.1}$$

When two nodes come into conflict with each other, a collision may happen causing node crashes with a probability p_c. Throughout this chapter, we consider two possible *scenarios*. The first corresponds to the case where at least one of the nodes is equipped with *perfect* detection and avoidance capabilities precluding the nodes from crashes ($p_c = 0$) by taking successful emergency avoidance actions to escape imminent crashes. This is indeed achieved at the cost of additional state information at nodes. The second corresponds to the occurrence of a collision causing node crash with probability one ($p_c = 1$). This occurs when no one of the mobile nodes is equipped with conflict detection and avoidance capabilities. Thus, in this scenario, a conflict occurrence leads to a systematic collision.

5.2.3 Collision-free conflict scenario

This scenario assumes that the mobile nodes are able to perfectly avoid imminent crashes.

Analysis

We describe here a stochastic model based on continuous-time Markov chains in order to study the conflicts incurred by opportunistic contacts between nodes. We are particularly interested in capturing how the network evolves in the presence of perfect conflict detection and avoidance capabilities. Under such conditions, there is no possibility that a pair of nodes is moved out of the network when an opportunistic conflict occurs, that is, there is no nodes loss due to conflicts. However, due to limited energy sources that can ensure continuous power harvesting, mobile nodes mission duration is limited: investing more energy in conflict detection and avoidance operations can increase the safety but decrease the nodes mission duration. Concepts to harmonize these tradeoffs are required, which means, among other concepts, that the number of conflicts encountered during a mission duration should be limited. Under such observations, a key issue is the resolution of the following question: how likely does the system evolve in the future under some initial

conditions? In particular, we are interested in quantitative estimate of the incurred cost due to the conflict detection and avoidance actions. Answering this question consists in finding appropriate state variables that describe the system evolution.

To this end, we look at two stochastic variables: the aggregate and the individual conflict numbers. Indeed, the energy consumption is directly related to the number of conflicts. The higher is the number of conflicts to resolve, the higher is the required energy for conflict detection and resolution manoeuvres. The *aggregate conflict number*, denoted by $n_a(t)$, is the number of *all* conflicts measured in the network between *any* two nodes until a given time t. The *individual conflict number*, denoted by $n_i(t)$, is the number of conflicts that an *arbitrary* node undergoes with any other node in the network until a given time t. From a performance viewpoint, $n_a(t)$ is a measure of the performance of the system as a whole, while $n_i(t)$ is a measure of the system performance as seen from an individual node.

Proposition 7. *Both aggregate and individual conflict numbers $n_a(\cdot)$ and $n_i(\cdot)$ obey Poisson processes with rates*

$$\lambda_a = \frac{N(N-1)}{2}\,\lambda \tag{5.2}$$

and

$$\lambda_i = (N-1)\,\lambda \tag{5.3}$$

respectively.

Proof. Assume that the process $n_a(\cdot)$ is at state j at time t, that is, $n_a(t) = j$. There are $\frac{N(N-1)}{2}$ different combinations for choosing pairs of nodes likely to come into conflict. Therefore, considering the exponential assumption of the inter-contact time, we have the same set of independent Poisson processes with rate λ running, such that the first one to go off determines the process $n_a(\cdot)$ to jump in state $j+1$. Recalling that merging independent Poisson processes produces a Poisson process whose rate is the sum of rates of individual ones, we find that the opportunities of conflicts between nodes, that is, $n_a(\cdot)$ can be represented by Poisson process having rate $\frac{N(N-1)}{2}\,\lambda$. Finally, by considering the fact that there are $(N-1)$ nodes likely to come into conflict with a given node u_i and applying the same reasonings to process $n_i(\cdot)$, we establish Proposition 7. □

Let us now introduce two key performance metrics, namely the *aggregate* and *individual inter-conflict times*. The former, denoted by T_{act}, stands for the elapsed time between any two successive conflicts whereas the latter, denoted by T_{ict}, is the time that it takes for a given target node to come into conflict with another node during the system evolution.

Corollary 1. *Aggregate and individual inter-conflict times T_{act} and T_{ict} follow exponential distributions with means $\frac{1}{\lambda_a}$ and $\frac{1}{\lambda_i}$ respectively, that is*

$$Pr(T_{act} \leq t) = 1 - e^{-\frac{N(N-1)}{2}\lambda t} \tag{5.4}$$

$$Pr(T_{ict} \leq t) = 1 - e^{-(N-1)\lambda t} \tag{5.5}$$

Proof. This follows from the simple fact that the aggregate and individual numbers of conflicts occurring during the system evolution are Poisson processes with rates λ_a and λ_i respectively. □

Discussion

Some general remarks can be drawn from the previous analysis. From proposition 7, we can obtain closed-form expressions for the expected aggregate and individual number of conflicts that occur by time t:

$$\mathrm{E}(n_a(t)) = \frac{N(N-1)}{2}\,\lambda\,t \tag{5.6}$$

$$\mathrm{E}(n_i(t)) = (N-1)\,\lambda\,t \tag{5.7}$$

These two metrics are valuable indicators for the assessment of conflicts risks and hence whether a conflict detection and avoidance procedure is needed. For a given deployment characterized by its parameter λ, duration t and number of deployed nodes N, an increased total number of conflicts indicates the high need of a conflict detection and avoidance procedure in order to keep the nodes mission safe. On the other hand, a very low number of conflicts indicates that the use of such a procedure is more harmful (futile energy consumption) than beneficial.

Similarly, from Corollary 1 we can derive closed-form expressions for the mean aggregate and individual inter-conflict times:

$$\mathrm{E}(T_{act}) = \frac{2}{N(N-1)\lambda} \tag{5.8}$$

$$\mathrm{E}(T_{ict}) = \frac{1}{(N-1)\lambda} \tag{5.9}$$

Using these expressions, we can decide if the system performance is acceptable for a given number of nodes and contact rate. Besides, these expressions allow us to gain a

better understanding of the impact of these parameters on the performance metrics. For example, in Figure 5.1 we plot the mean aggregate inter-conflict time $\mathrm{E}(T_{act})$ as a function of the number of nodes for different contact rates, a small contact rate $\lambda = 0.01$, a medium contact rate $\lambda = 0.1$ and a large contact rate $\lambda = 1$. The contact rate $\lambda = 0.01$ corresponds to a sufficiently large deployment region where a single pair of deployed nodes rarely come in contact with each other; the contact rate $\lambda = 0.1$ corresponds to a medium deployment region where a single pair of nodes occasionally experience conflicts and the contact rate $\lambda = 1$ corresponds to a small deployment region where a single pair of nodes frequently come in contact with each other. This figure suggests that the impact of decreasing the contact rates on the mean aggregate inter-conflict time is significant in the case of low density of nodes while this impact becomes meaningless in the case of high density.

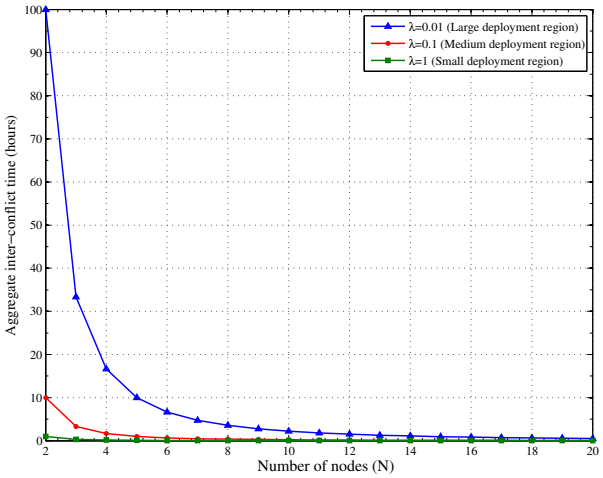


Figure 5.1 – Aggregate inter-conflict time for different contact rates

To assess the system safety degree, we define p_{sa} as the probability that the whole system survives until time t_{sa} without having encountered any conflict between pairs of nodes and p_{si} as the probability that an arbitrary node survives until time t_{si} without having encountered any conflict. In what follows, p_{sa} (resp. p_{si}) is called the *aggregate* (resp. the *individual*) *survival probability*; t_{sa} (resp. t_{si}) is called the *aggregate* (resp. the *individual*) *safety period*. Based on corollary 1, we obtain:

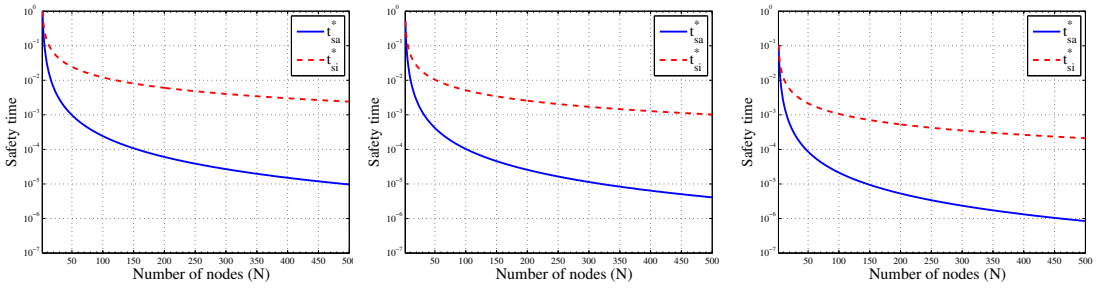
$$p_{sa} = \Pr(T_{act} > t_{sa}) = e^{-\frac{N(N-1)}{2}\lambda t_{sa}} \tag{5.10}$$

$$p_{si} = \Pr(T_{ict} > t_{si}) = e^{-(N-1)\lambda t_{si}} \tag{5.11}$$

Relations (5.10) and (5.11) are of great importance and enable us to derive the safety period for a given survival probability as a function of the number of nodes N and the pair-wise contact rate λ. More interestingly enough, they enable us to find the concentration

of nodes to be deployed while achieving a threshold safety period with high probability. However, to obtain general insights, it is more convenient to rescale both the aggregate and individual safety periods in order to construct two new dimensionless variables $t^*_{sa} = \lambda\, t_{sa}$ and $t^*_{si} = \lambda\, t_{si}$. The reason for this variable rescaling stems from the fact that $\frac{1}{\lambda}$, the mean pairwise contact time, is the natural time scale of the model. Therefore, t^*_{sa} (resp. t^*_{si}) is nothing but the ratio between the aggregate (resp. individual) safety time and the mean pairwise contact time.

We obtain then Figure 5.2, where the scaling variables $t^*_{sa} = \dfrac{-2\log(p_{sa})}{N(N-1)}$ and $t^*_{si} = \dfrac{-\log(p_{si})}{(N-1)}$ are plotted against the size of the network N for different survival probabilities $p_{sa} = p_{si} = 0.3$; $p_{sa} = p_{si} = 0.6$ and $p_{sa} = p_{si} = 0.9$. Many key features can be drawn from this figure. First, note that for a fixed value of N, the rescaled variables noticeably decrease as the survival probability increases. For example, for $N = 10$, the obtained aggregate safety time is about 2.67% of the mean contact time with $p_{sa} = 0.3$. However, it falls to 1.13% of the mean contact time with $p_{sa} = 0.6$ and to 0.23% of the mean contact time with $p_{sa} = 0.9$. Second, a closer inspection of this plot indicates that at fixed survival probability, the obtained safety times rapidly decrease to small values.



(a) $p_{sa} = p_{si} = 0.3$ (b) $p_{sa} = p_{si} = 0.6$ (c) $p_{sa} = p_{si} = 0.9$

Figure 5.2 – Safety periods for different survival probabilities

5.2.4 Collision-prone conflict scenario

This scenario assumes that nodes take no avoidance action whatsoever in response to imminent crashes. This can particularly be the case for small mobile nodes that are unable to meet conflict detection and resolution equipment requirements due to their limited size, weight and energy.

Analysis

Next, we define the partial survival time T_{2i} as the time till the i^{th} collision.

Proposition 8. *Under the conflict with collision model, the Laplace transform of the partial survival time can be expressed as*

$$\widehat{T}_{2i}(\theta) = \prod_{j=0}^{i-1} \frac{\lambda \, \mathsf{C}_{N-2j}^2}{\theta + \lambda \, \mathsf{C}_{N-2j}^2} \tag{5.12}$$

Proof. The proof is based on modeling the number of nodes that are removed from the system until time t due to collision occurrences as a discrete time Markov chain. Since we assumed that a conflict results in a loss of a node pair from the system, the states of the Markov chain take only even values $k = 2i$ for $i = 0, 1, \cdots, \lfloor \frac{N}{2} \rfloor$, where the $\lfloor \cdot \rfloor$ symbol stands for the floor function. Note that all states are transient except state $2\lfloor \frac{N}{2} \rfloor$ which is an absorbing state. The transition diagram of the Markov chain is given in Figure 5.3. When this chain is at state $2i$ the system contains $N - 2i$ nodes still alive, there are hence C_{N-2i}^2 independent exponential clocks with rate λ running, such that the first one to go off triggers a collision between two nodes, thereby yielding a jump to state $2(i+1)$. Therefore, the transition from state $2i$ to state $2(i+1)$ occurs at the rate $\lambda \mathsf{C}_{N-2i}^2$ for $i = 0, 1, \cdots, \lfloor \frac{N}{2} \rfloor - 1$. If we let S_{2i} to be the sojourn time at state $2i$ for $i = 0, 1, \cdots, \lfloor \frac{N}{2} \rfloor - 1$, then S_{2i} is exponentially distributed with intensity $\lambda \mathsf{C}_{N-2i}^2$. Moreover, S_{2i} are mutually independent random variables. The partial survival time T_{2i} can be then expressed as

$$T_{2i} = \sum_{j=0}^{i-1} S_{2j} \tag{5.13}$$

Recalling that the Laplace transform of the sum of mutually independent random variables corresponds to the multiplication of their Laplace transforms and that the Laplace transform of the exponentially distributed random variable S_{2j} is $\widehat{S}_{2j}(\theta) = \frac{\lambda \, \mathsf{C}_{N-2j}^2}{\theta + \lambda \, \mathsf{C}_{N-2j}^2}$, we obtain (5.12). □

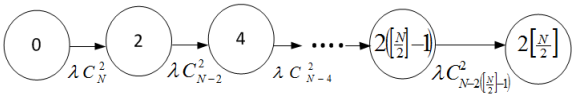


Figure 5.3 – Transition diagram of the Markov chain for the number of nodes removed from the system

Proposition 9. *Under the conflict model with collisions, the expected partial survival time is given by:*

$$\boldsymbol{E}(T_{2i}) = \frac{2}{\lambda} \sum_{j=0}^{2i-1} \frac{(-1)^{j+1}}{N-j} \tag{5.14}$$

The asymptotic expansion of $\boldsymbol{E}(T_{2i})$ for a large number of nodes N is:

$$\boldsymbol{E}(T_{2i}) = o(1) \tag{5.15}$$

Proof. From (5.13), we obtain

$$\begin{aligned}
\mathbf{E}(T_{2i}) &= \sum_{j=0}^{i-1} \mathbf{E}(S_{2j}) \\
&= \sum_{j=0}^{i-1} \frac{1}{\lambda \mathsf{C}_{N-2j}^2} \\
&= \frac{2}{\lambda} \sum_{j=0}^{i-1} \frac{1}{(N-2j)(N-2j-1)} \\
&= \frac{2}{\lambda} \sum_{j=0}^{i-1} \left\{ \frac{1}{N-2j-1} - \frac{1}{N-2j} \right\} \\
&= \frac{2}{\lambda} \sum_{j=0}^{2i-1} \frac{(-1)^{j+1}}{N-j}
\end{aligned}$$

For sake of clarity the proof of the asymptotic expansion of $\mathbf{E}(T_{2i})$ is forwarded to the appendix. □

Discussion

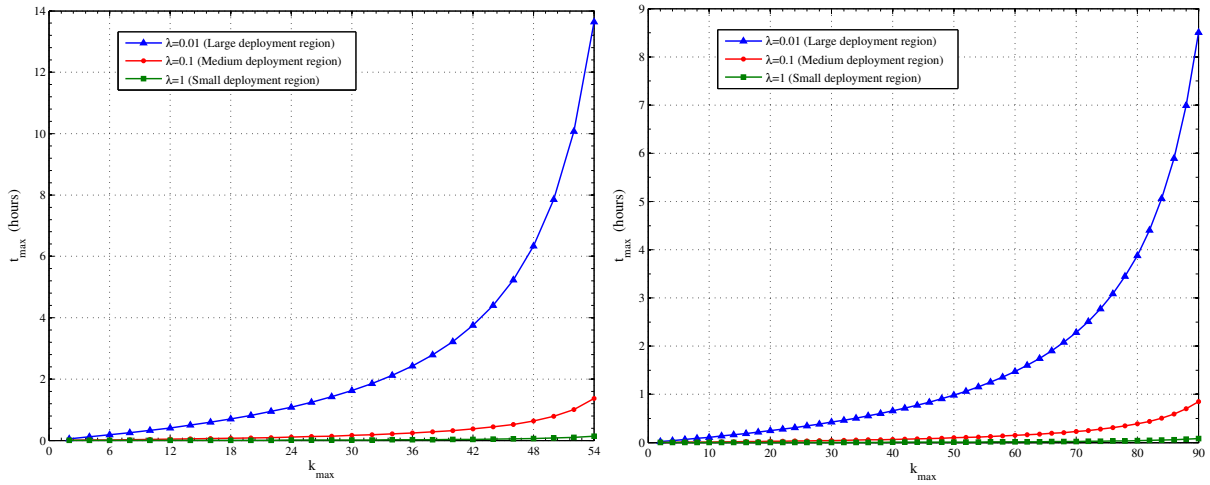
The collision-prone conflict scenario assumes that nodes are not equipped with conflict detection and avoidance capabilities and consequently amounts to the systematic loss of a node pair as a result of a conflict. In this case, the user may tolerate collisions up to a certain level. Such tolerance can be considered in missions with redundant and cheap nodes.

If we note $k_{max} = 2\, i_{max}$, the maximum number of mobile nodes that are tolerated to collide, then the maximum allowed mission duration t_{max} can be estimated using equation (5.14) as follows:

$$t_{max} = \frac{2}{\lambda} \sum_{j=0}^{k_{max}-1} \frac{(-1)^{j+1}}{N-j} \tag{5.16}$$

In Figure 5.4, we plot the maximum allowed mission duration t_{max} while varying the maximum number of nodes that are tolerated to collide k_{max}. We consider two numbers of deployed nodes $N = 60$ (Figure 5.4(a)) and $N = 100$ (Figure 5.4(b)) and different contact rates $\lambda = 0.01$, $\lambda = 0.1$ and $\lambda = 1$. For $N = 60$, k_{max} is varied from 2 to 54, that is the number of nodes that are tolerated to be lost is varied from 3.33% to 90%

of the total number of deployed nodes. For $N = 100$, k_{max} is varied from 2 to 90, that is the number of nodes that are tolerated to be lost is varied from 2% to 90% of the total number of deployed nodes. As shown in this figure, the maximum allowed mission duration t_{max} increases as λ and k_{max} increase and decreases as N increases. For example with $N = 60$, 90% ($k_{max} = 54$) of the deployed nodes will be lost within less than 2 hours with a contact rate $\lambda = 0.1$, while with $\lambda = 0.01$ this duration increases to 13.5 hours.



(a) $N = 60$ (b) $N = 100$

Figure 5.4 – Maximum allowed mission duration

5.3 Simulation Study

We perform in this section a serie of simulations to validate the obtained analytical results. We particularly apply our analytical model to a swarm of small aerial vehicles moving in a 3D confined airspace. We use four simulation procedures. The first procedure aims at validating the fundamental hypothesis about the exponential distribution of the pairwise inter-contact time under two mobility models, namely the random waypoint and the random direction mobility models. The second and the third procedures aim respectively at the validation of the analytical results on the collision-free conflict scenario and the collision-prone conflict scenario. We finally provide in the last subsection an explicit expression of the parameter λ and confirm the obtained expression using simulation results.

5.3.1 Simulation setup

Although the expressions obtained in section 5.2 hold for all mobility models with exponential inter-contact times, next we shall only apply them to two standard mobility mod-

els, namely the random waypoint (RWP) and the random direction (RD) model [Roy 10]. There are several versions of these two mobility models. In this work, we consider the 3D models with no pause times. In the RWP, each node is assigned an initial position and moves linearly with a constant speed to a destination waypoint chosen uniformly in the 3D simulation region. Upon reaching the destination waypoint, a new waypoint is chosen independently of all previous waypoints. This is iterated until the end of the simulation. In the RD mobility, mobile nodes choose a destination direction, chosen from a uniform distribution, rather than a destination point. The node linearly moves in this direction till it reaches the 3D simulation region boundary and then reflects back.

The results obtained hereafter are based on simulations using the OMNET++ simulator [Varg 08] where nodes are moving according to the specified mobility models with a constant velocity $v = 5m/s$ in a mall of 500m $\times$ 400m $\times$ 30m.

In all our experiments, we run as many trials as needed in order to reach a 95% confidence interval

5.3.2 Validation of the exponential property

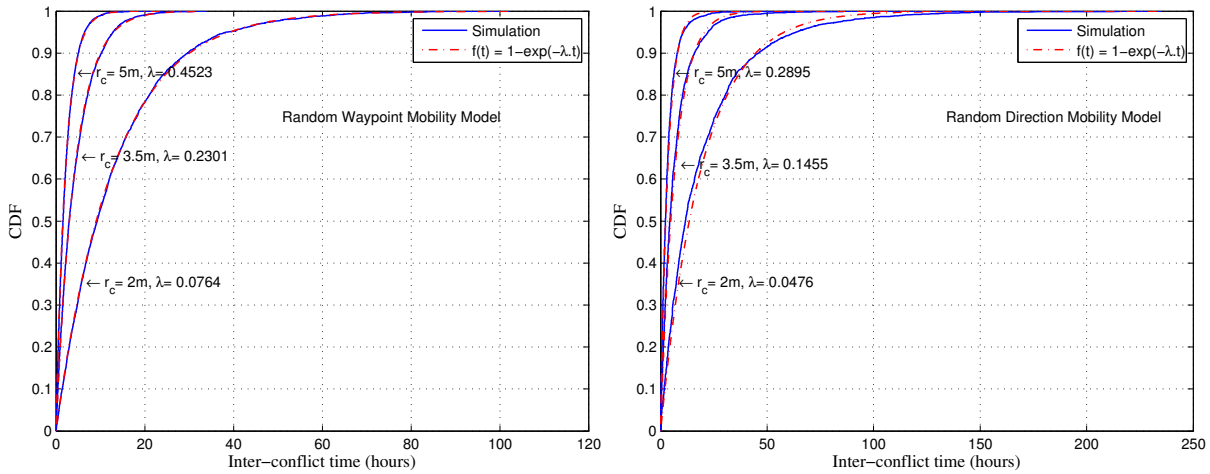
In order to apply the results in section 5.2, we need first to check the validity of the exponential distribution of the pairwise inter-contact time under the two mobility models and to estimate the parameter λ.

For each mobility model and for various contact ranges, we have simulated the movement of two nodes and recorded their inter-contact times. The obtained results are illustrated in Figure 5.5, where the CDF of the inter-contact time between the two nodes is depicted for three different contact ranges (r_c =2m, 3.5m, 5m). we also plot the CDF of an exponential distribution with intensity λ. For the two mobility models and for each contact range, the value of the corresponding parameter λ is estimated as the inverse of the average inter-contact times obtained across all the experiment repetitions. The estimated contact rates λ are expressed in hour^{-1}. We observe a good matching between the distribution of the recorded inter-contact times (solid lines) and the exponential distribution (dashed lines).

The values of the parameter λ of the pairwise inter-contact distribution is evaluated, we can now verify the correctness of our analytical results.

5.3.3 Validation of the collision-free conflict scenario

For the two mobility models and for the three contact ranges (r_c =2m, 3.5m, 5m), we plot in Figure 5.6 (resp. in Figure 5.7) the aggregate (resp. the individual) number of conflicts that occur during an hour of simulation under varying number of nodes. The analytical



(a) Random Waypoint (b) Random Direction

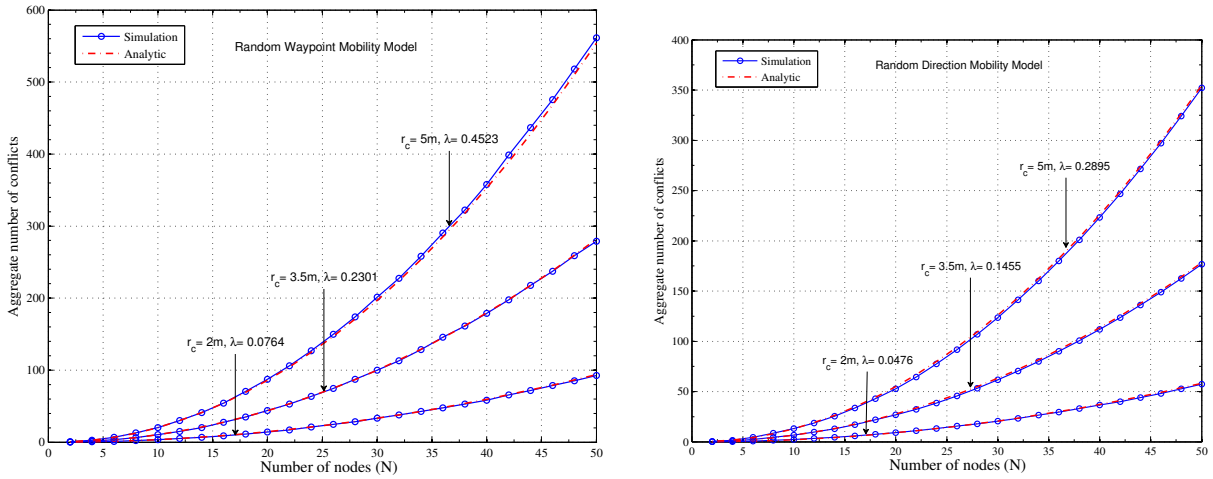
Figure 5.5 – CDF of the inter-conflict time between two nodes

results (dashed lines) for these two metrics are calculated according to equations (5.6) and (5.7). We can see an excellent agreement between the analytical and the simulation results. These results prove hence the ability of our analytical model to accurately predict the aggregate (resp. the individual) number of conflicts under different mobility patterns, contact ranges and number of nodes.

Figure 5.6 (resp. Figure 5.7) also show that the aggregate (resp. the individual) number of conflicts increases as the contact range r_c increases. For example with the RWP mobility and for $N = 30$, the total number of aggregate conflicts is of around 33 conflicts with a contact range $r_c = 2m$ and it increases to 100 conflicts with $r_c = 3.5m$ and to 197 conflicts with $r_c = 5m$. This is because the larger is the contact range, the more crowded will be the simulation environment (with respect to r_c) and consequently more frequently will the nodes meet.

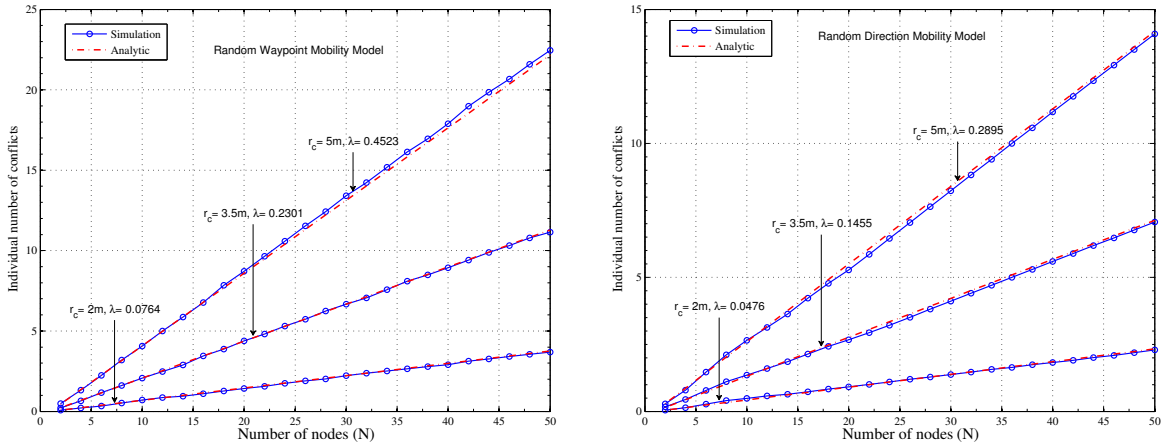
Figure 5.8 (resp. Figure 5.9) displays the mean aggregate (resp. individual) inter-conflict time on a log-scale, obtained both through simulations and by the analytical model, as a function of an increasing number of nodes. Once again, we can see that the experimental results are conform to the analytical values that we obtain from equations (5.8) and (5.9). We can as well see that the aggregate (resp. the individual) inter-conflict time decreases as the contact range r_c increases. Indeed the larger is the contact range, the more often will the nodes meet and hence the shorter will be the inter-conflict times.

Figure 5.10 (resp. Figure 5.11) plots the experimental and the analytical results for the aggregate (resp. the individual) survival probability with a number of nodes $N = 20$. The analytical results are calculated according to equations (5.10) and (5.11) and are shown to be in a quite good fit with the experimental results. Without surprise, these two figures indicate that the aggregate and the individual survival probabilities decrease



(a) Random Waypoint (b) Random Direction

Figure 5.6 – Aggregate number of conflicts under different numbers of nodes



(a) Random Waypoint (b) Random Direction

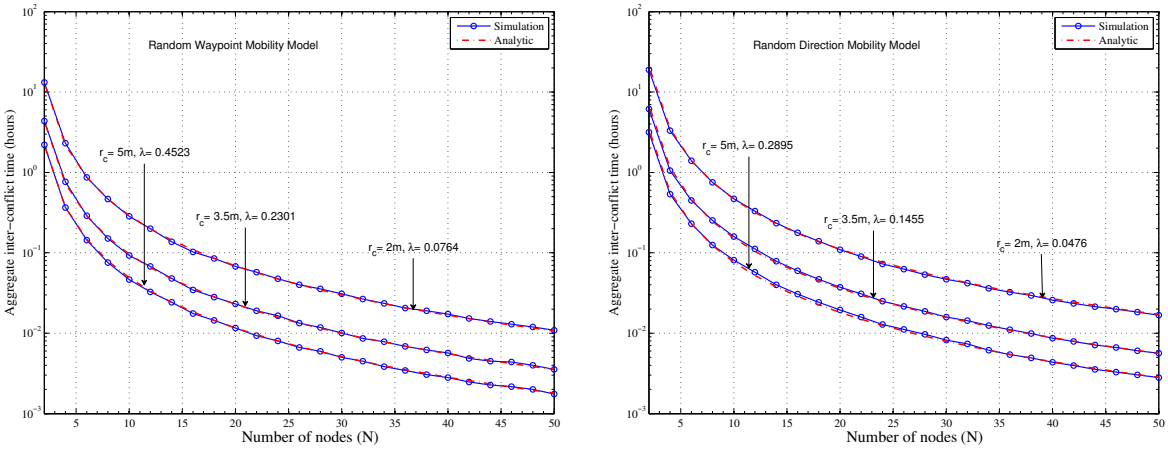
Figure 5.7 – Individual number of conflicts under different numbers of nodes

as the survival times and the contact ranges r_c increase.

5.3.4 Validation of the collision-prone conflict scenario

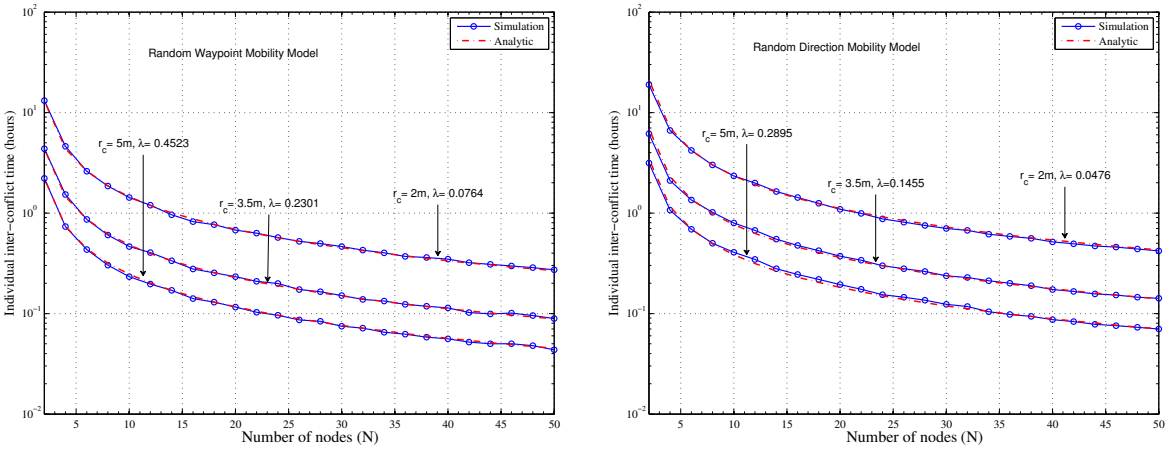
In Figure 5.12 and Figure 5.13, we validate the analytical results obtained for the collision-prone conflict scenario by comparing them against the simulation results. The represented analytical results are calculated according to equations (5.14).

For each mobility model and contact range, we plot in Figure 5.12 the expected partial survival time under varying number of nodes. In Figure 5.12 (a) and (b) we report, on a log-scale for the y axis, the partial survival time $\mathbf{E}(T_{10})$, that is the survival time till state $k = 2i = 10$. In Figure 5.12 (c) and (d) we report the partial survival time $\mathbf{E}(T_{20})$.



(a) Random Waypoint (b) Random Direction

Figure 5.8 – Aggregate inter-conflict time under different numbers of nodes

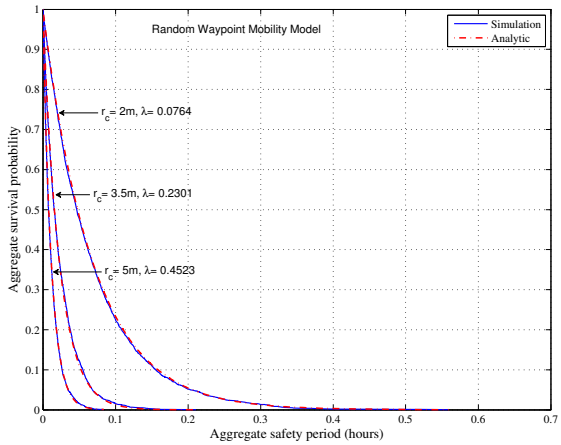


(a) Random Waypoint (b) Random Direction

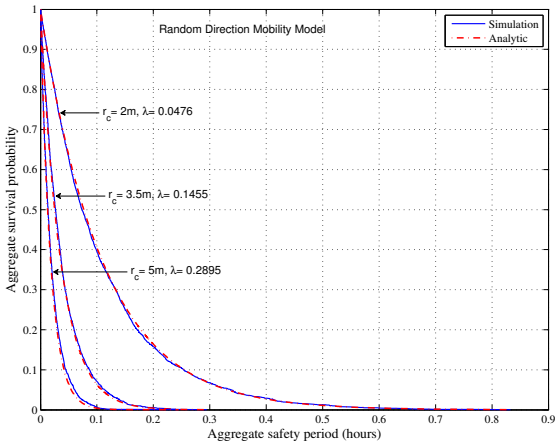
Figure 5.9 – Individual inter-conflict time under different numbers of nodes

We observe a close fit between the analytical and the simulation results under all of the considered contact ranges, number of nodes, mobility models and states. We can also see, with no surprise, that the partial survival time decreases as the contact range and the number of nodes increases.

Figure 5.13 plots the partial survival time of the system for a fixed number of nodes $N = 40$ and throughout the different states $k = 2i$, for $i = 1, \cdots, \lfloor \frac{N}{2} \rfloor$. We can similarly see a good fit between the analytical and the simulation results. These results give hence a good indication that our model, despite its simplicity, is able to accurately estimate the system evolution over its different states.

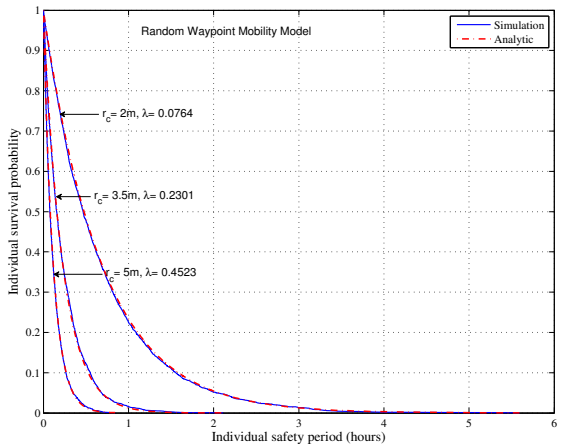


(a) Random Waypoint

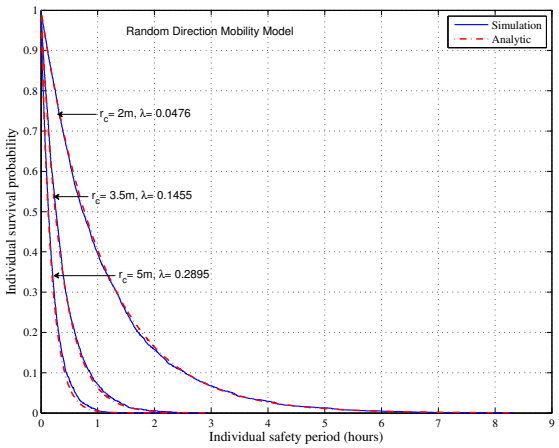


(b) Random Direction

Figure 5.10 – Aggregate survival probability with a number of nodes $N = 20$



(a) Random Waypoint



(b) Random Direction

Figure 5.11 – Individual survival probability with a number of nodes $N = 20$

5.3.5 Estimates for the parameter λ

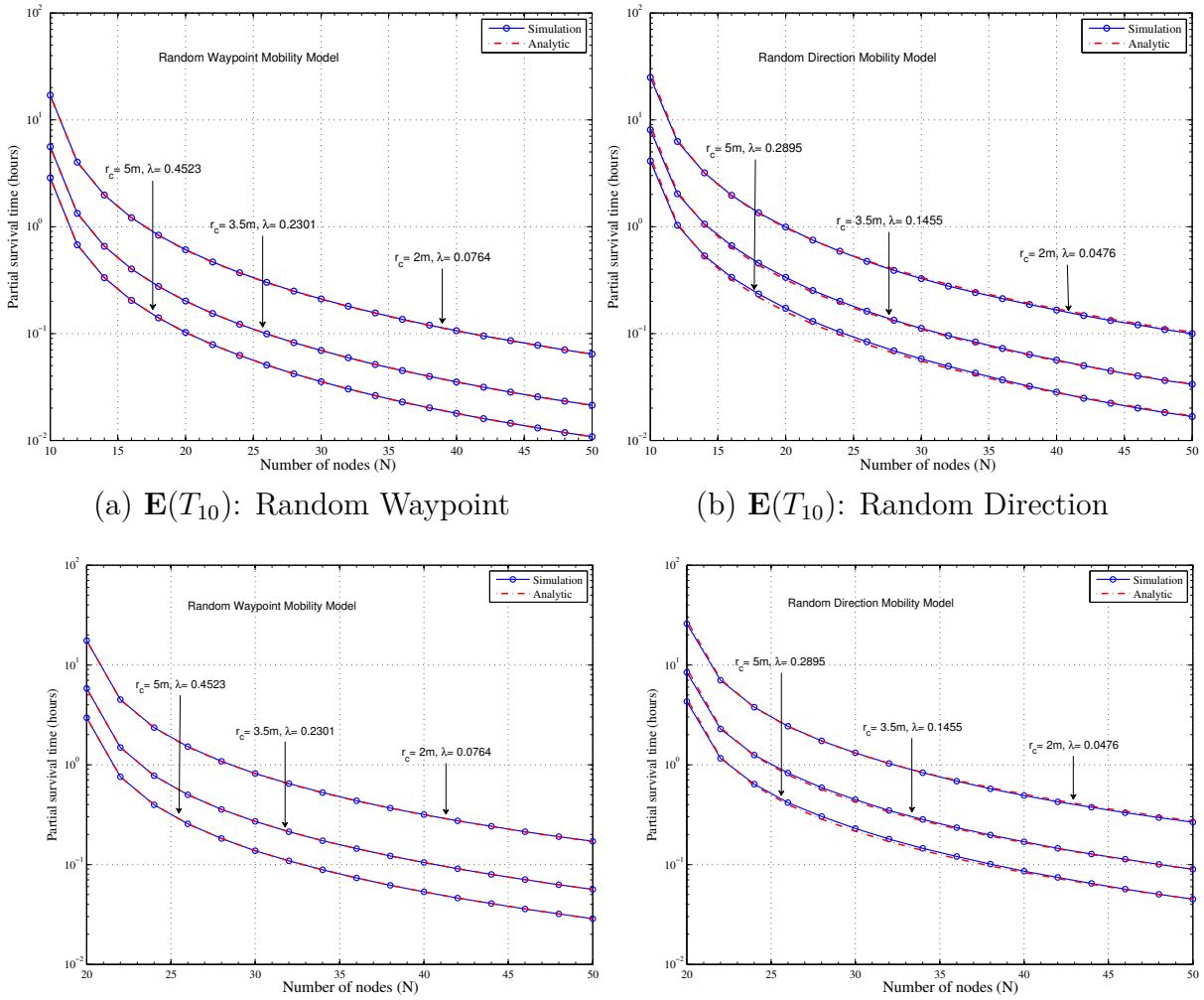
In this subsection, we give an explicit expression of the parameter λ of the exponential distribution and validate the obtained expression against simulation results.

Proposition 10. *Under mobility models with uniform node spacial distribution, the parameter λ is given by:*

$$\frac{\pi\, r_c^2\, v_r}{V} \tag{5.17}$$

where v_r is the mean relative velocity between the nodes and V is the deployment region volume.

Proof. To determine the expression of the parameter λ, we will first determine the expres-



(a) $\mathbf{E}(T_{10})$: Random Waypoint

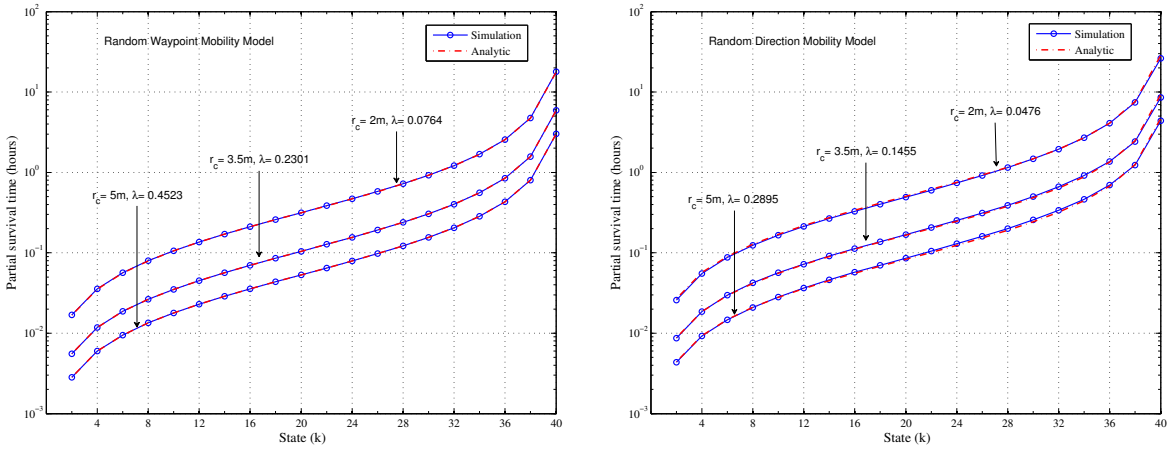
(b) $\mathbf{E}(T_{10})$: Random Direction

(c) $\mathbf{E}(T_{20})$: Random Waypoint

(d) $\mathbf{E}(T_{20})$: Random Direction

Figure 5.12 – Partial survival time under different numbers of nodes

sion of the number of conflicts that a given node A undergoes per an interval of time Δt. As shown in Figure 5.14, a conflict occurs when the centres of two nodes are separated by a distance less or equal to the contact range r_c. Assuming that all other nodes except node A are in stationary positions, we can see that node A will be in conflict only with nodes whose centres are within the conflict cylinder depicted in Figure 5.14. Without much impact, we can assume that node A has no direction changes during Δt. In the time interval Δt, node A will move a distance $v_a \, \Delta t$ represented by the length of the cylinder (v_a is the average velocity of node A). The number of conflicts node A undergoes in the time interval Δt will therefore be equal to the density of the surrounding nodes $\rho = \frac{N-1}{V}$ (uniform density) multiplied by the volume $\pi \, r_c^2 \, v_a \, \Delta t$ of the conflict cylinder. Since the other nodes are not really stationary, we have to replace v_a by the mean relative



(a) Random Waypoint (b) Random Direction

Figure 5.13 – The system evolution under a number of nodes $N = 40$

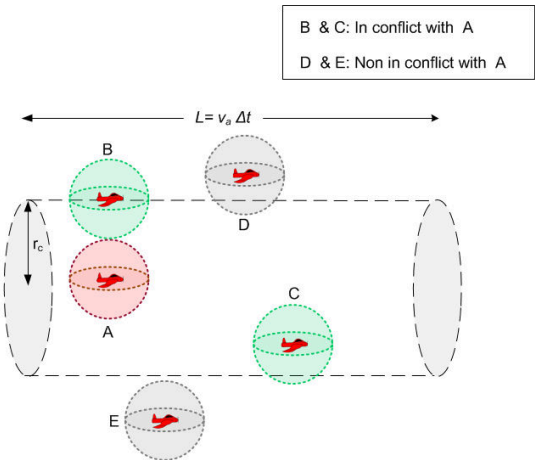


Figure 5.14 – Conflict cylinder

velocity v_r. The number of conflicts is therefore equal to:

$$(N-1)\frac{\pi\, r_c^2\, v_r}{V}\Delta t \tag{5.18}$$

By comparing equation (5.18) and equation (5.7) that estimates the number of individual conflicts as a function of λ, the parameter λ can finally be identified as $\frac{\pi\, r_c^2\, v_r}{V}$. □

Many system performance related questions can be answered by combining proposition 10 and the previously estimated metrics. For example, which contact range (and hence which node size) should be used to ensure a given safety period? Which deployment region size should be used in order to guarantee a conflict free mission?

For both of the considered mobility models RWP and RD, if the nodes are traveling at the same constant speed v then the mean relative speed v_r can be expressed as a

proportional function of v as follows [Groe 05]: $v_r = \frac{4}{\pi} v$. As for the nodes distribution, it is known that the nodes spacial distribution in the RD mobility model is uniform [Nain 05]. The expression of λ under the RD mobility with constant velocities is hence:

$$\lambda_{RD} = \frac{4\, r_c^2\, v}{V} \tag{5.19}$$

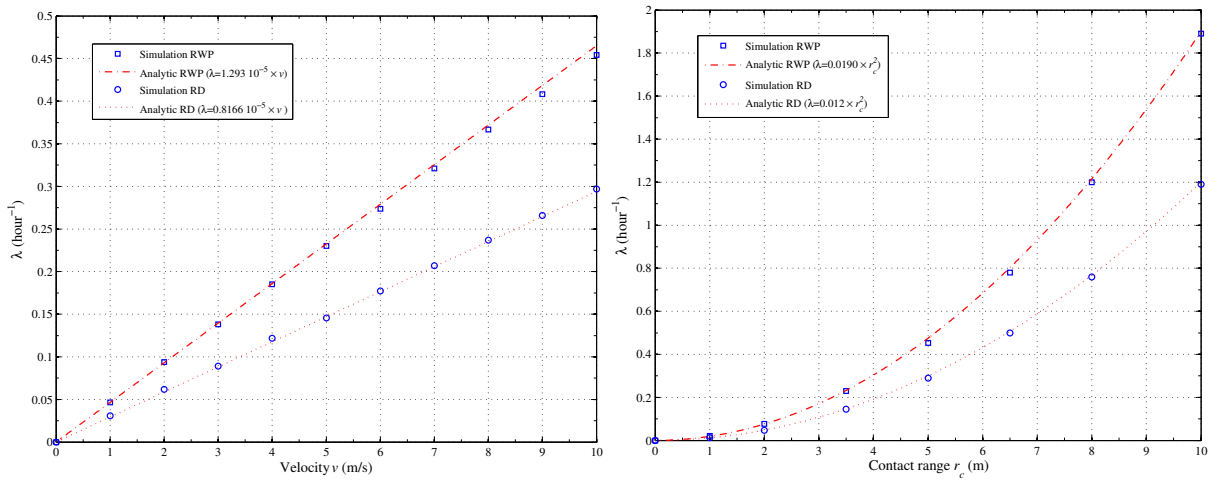
This is in contrast with the RWP mobility model where it has been observed that nodes are more concentrated around the centre of the deployment region [Lass 06]. In this case the density is equal to $(N-1)\,\gamma^*$, where γ^* depends on the deployment region. Explicit expressions for γ^* over any convex region, are given in [Lass 06] and [Hyyt 05]. The parameter λ under the RWP mobility with constant velocities can consequently be expressed as:

$$\lambda_{RWP} = 4\,\gamma^*\, r_c^2\, v \tag{5.20}$$

It is worthy of attention in equations (5.19) and (5.20) that λ has a linear relationship with the velocity v and a square linear relationship with the contact range r_c. This is confirmed in Figure 5.15 where the analytic estimates of λ based on equations (5.19) and (5.20) are plotted against the velocity (Figure 5.15(a)) and the contact range (Figure 5.15(b)). In Figure 5.15(a) the contact range is fixed to $r_c = 3.5m$, we hence have $\lambda_{RD} = \frac{4\, r_c^2}{V}\, v = 0.8166\, 10^{-5}\, v$ and $\lambda_{RWP} = 4\,\gamma^*\, r_c^2\, v = 1.293\, 10^{-5}\, v$ ($\gamma^* = 0.2639\, 10^{-6}$). In Figure 5.15(b) the velocity is fixed to $v = 5m/s$, we hence have $\lambda_{RD} = \frac{4\, v}{V}\, r_c^2 = 0.012\, r_c^2$ and $\lambda_{RWP} = 4\,\gamma^*\, v\, r_c^2 = 0.019\, r_c^2$. The contact rate λ is expressed in hour^{-1}. Not surprising, the value of λ is higher for nodes moving according to the RWP mobility model than for the RD mobility since the nodes are more concentrated near the centre of the region. We also plot the values of λ obtained through simulations and observe a good matching between the analytic and the simulation based results.

5.4 Conclusion

In this chapter, we have introduced a simple, yet accurate stochastic model with only two input parameters to characterize conflicts in a swarm of 3D-mobile nodes sharing the same airspace. We have considered two scenarios: a scenario in which nodes have perfect detection and avoidance capabilities and an other scenario where nodes have no detection and avoidance capabilities. Closed-form expressions were obtained for many conflict related metrics such as the aggregate and individual safety periods, survival probabilities and number of conflicts. These analytical results were compared to simulation results obtained under two mobility models: the random waypoint and the random direction mo-



(a) (b)

Figure 5.15 – Relationship between the parameter λ and (a) the velocity v (b) the contact range r_c

bility models. The analytical results were in close matching with the simulation results which is a good indication that our model, in spite of its simplicity, is able to accurately assess conflict risks.

Chapter 6

Summary and Perspectives

Next, we will summarize the contributions of this thesis and bring up ideas and directions to expand and improve these contributions.

6.1 Summary

In this thesis, we treated two fundamental problems in mobile wireless networks: localization and conflict detection. Localization refers to the determination of the coordinates of the mobile nodes. Conflict detection refers to the detection of conflict situations between the mobile nodes. A conflict is defined as an event in which the distance between two nodes breaks the minimal defined separation criterion. The contributions of this thesis are the following.

- Localization:
  - we proposed a distributed localization algorithm for mobile wireless networks. Our algorithm overcomes the limitations of previous localization techniques in that it: 1) requires no synchronization between the nodes 2) makes no assumption about the nodes transmission range 3) aggregates different instead of a single metric in order to weight the collected location information guaranteeing as such a good robustness against the uncertainty of certain metrics 4) is highly configurable through a set of user-specifiable parameters. For instance the algorithm can be range-free or range-based depending of the availability of the range information, can use only anchor nodes location information or both anchor and normal nodes location information, etc.
- Conflict detection:

- Our first contribution in the context of conflict detection consists in a spatio-temporal conflict detection approach for 3D-mobile communicating nodes. We first provided a comprehensive mathematical framework for 3D conflict detection. Based on this framework, we proposed SLIDE a new distributed and straight line conflict detection and alerting algorithm. SLIDE has two major advantages over the previously proposed conflict detection algorithms: 1) it does not require the mobile nodes predetermined trajectories 2) it considers both imperfect sensing capabilities and communication links providing hence a better efficiency in real world applications.
- In the same context, we proposed a stochastic model evaluating the conflict risks in a swarm of 3D-mobile nodes sharing the same airspace. Our model provides closed-form expressions for various conflict related metrics based only on two input parameters, namely the number of nodes N and a parameter λ characterizing the inter-contact time between a pair of mobile nodes. The proposed stochastic model can be a useful tool for the swarm designer. It indeed assists him/her in 1) taking well-founded and safe (with reduced conflict risks) design decisions 2) understanding the mobile nodes system evolution and 3) answering many questions regarding the conflict risks.

6.2 Perspectives

The work presented in this thesis opens up many new possibilities for future work. Our future work can be summarized in the following points:

1. The performance of our localization algorithm depends on different parameters. We have provided theoretical recommendations on the appropriate setting of these parameters. However, theoretical recommendations do not always reflect the practical situations. Besides, some tradeoffs need to be deeply analyzed in order to determine the appropriate optimal setting. For instance, decreasing the broadcast period T_a would generally improve the algorithm accuracy but on the cost of higher communication overhead. There is nevertheless an optimal point after which decreasing T_a will not have a positive effect on the accuracy of the algorithm. Values lower than the optimal value unnecessarily increase the communication cost. In a very near future, we will comprehensively study the performance of our algorithm under various values of its parameters. This study will enable us draw practical guidelines on how to set these parameters in order to reach the best performance of the algorithm.

2. The performance of our conflict detection algorithm highly depends on the setting

of its two tuning parameters: the broadcast cycle T_b and the look ahead time T_l describing the time within which a given mobile node will have a straight line trajectory. We have provided a mathematical expression for the setting of the broadcast cycle T_b. This expression was validated using simulations. We were, nevertheless, limited to some intuitive directives for the setting of the look ahead time T_l. We also used a fixed look ahead time during our simulations. In a future work, we aim to use a dynamic look ahead time based on a prediction model. In particular, before projecting its current states to detect future conflicts, a mobile node will use its previous mobility traces and an adequate prediction model to dynamically estimate the look ahead time that will be used for the current projection.

3. To validate the analytical results of our stochastic model, we used simulations under two artificial mobility models, namely the random waypoint and the random direction mobility models. Simulation results are found to be in a good agreement with the analytical results. These results offered us a useful first insight about the accuracy of our model. However, for a better reliability, we plan to further validate our analytical model using real world mobility traces. We expect to obtain good results but obviously not as good as those obtained under the artificial models.

4. Once these points are dealt with, we intent to combine our three contributions into a single integrated framework for both localization and conflict detection. In our conflict detection algorithm SLIDE, we have assumed that the mobile nodes acquire their coordinates using an embedded GPS. In the new framework, nodes will instead use our proposed localization algorithm to determine their coordinates. The output of the localization algorithm will be used as an input for the conflict detection. Besides, we will combine the spatio-temporal conflict detection with the stochastic conflict analysis to achieve a more reliable and effective conflict detection.

Appendix

1 Asymptotic Expansion of $G_{n,i}$

Let us define the following series

$$G_{n,i} = \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{n-i+1} \tag{1}$$

where n and i are non-negative integer such as $1 < i < n$. We can show that $G_{n,i}$ has the following asymptotic expansion as long as i is held small and $n \to +\infty$

$$G_{n,i} = \ln(\frac{n}{n-i}) + o(1). \tag{2}$$

For this purpose, remark that $G_{n,i}$ can be expressed as

$$G_{n,i} = H_n - H_{n-i} \tag{3}$$

where H_n is the Harmonic series defined as

$$H_n = \sum_{k=1}^{n} \frac{1}{k}.$$

According to [Sedg 13], H_n has the following asymptotic expansion as $n \to +\infty$

$$H_n = \ln(n) + \gamma + o(1) \tag{4}$$

where γ is the Euler's constant. It follows that

$$H_{n-i} = \ln(n-i) + \gamma + o(1) \tag{5}$$

by plugging (4) and (5) into (3), we obtain (2).

2 Asymptotic Expansion of $\mathbf{E}(T_{2i})$

In this Appendix, we will show that $\mathbf{E}(T_{2i})$ has the following asymptotic expansion as $N \to \infty$

$$\mathbf{E}(T_{2i}) = o(1) \tag{6}$$

This asymptotic expansion is obtained by rewriting $\mathbf{E}(T_{2i})$ as

$$\mathbf{E}(T_{2i}) = \frac{2}{\lambda}\Big(Q_1(N) - Q_2(N)\Big) \tag{7}$$

where

$$Q_1(N) = \sum_{j=0}^{i-1} \frac{1}{N-2j-1} \text{ and } Q_2(N) = \sum_{j=0}^{i-1} \frac{1}{N-2j}. \tag{8}$$

We can remark that

$$Q_2(N) = \frac{1}{2}\Big(H_{\frac{N}{2}} - H_{\frac{N}{2}-i}\Big) \tag{9}$$

And

$$Q_1(N) + Q_2(N) = H_N - H_{N-2i} \tag{10}$$

Combining (9) and (10) into (7), we finally obtain

$$\begin{aligned} \mathbf{E}(T_{2i}) &= \frac{2}{\lambda}\Big\{(H_N - H_{N-2i}) - (H_{\frac{N}{2}} - H_{\frac{N}{2}-i})\Big\} \\ &= o(1) \end{aligned}$$

Bibliography

[Abu 16] Z. Abu-Shaban, X. Zhou, and T. D. Abhayapala. "A novel TOA-based mobile localization technique under mixed LOS/NLOS conditions for cellular networks". *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 11, pp. 8841–8853, 2016.

[Aite 03] E. Aitenbichler and M. Muhlhauser. "An IR local positioning system for smart items and devices". In: *23rd International Conference on Distributed Computing Systems Workshops*, pp. 334–339, IEEE, 2003.

[Alam 09] S. Alam, K. Shafi, H. A. Abbass, and M. Barlow. "An ensemble approach for conflict detection in Free Flight by data mining". *Transportation research part C: emerging technologies*, Vol. 17, No. 3, pp. 298–317, 2009.

[Alba 09] B. Albaker and N. Rahim. "Straight projection conflict detection and cooperative avoidance for autonomous unmanned aircraft systems". In: *4th IEEE Conference on Industrial Electronics and Applications*, pp. 1965–1969, IEEE, 2009.

[Alej 09] D. Alejo, R. Conde, J. Cobano, and A. Ollero. "Multi-UAV collision avoidance with separation assurance under uncertainties". In: *IEEE International Conference on Mechatronics*, pp. 1–6, IEEE, 2009.

[Alej 13] D. Alejo, J. M. Díaz-Báñez, J. A. Cobano, P. Pérez-Lantero, and A. Ollero. "The velocity assignment problem for conflict resolution with multiple aerial vehicles sharing airspace". *Journal of Intelligent & Robotic Systems*, Vol. 69, No. 1-4, pp. 331–346, 2013.

[Arch 08] J. K. Archibald, J. C. Hill, N. A. Jepsen, W. C. Stirling, and R. L. Frost. "A satisficing approach to aircraft conflict resolution". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 38, No. 4, pp. 510–521, 2008.

[Arth 15] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. "Instant outdoor localization and slam initialization from 2.5 d maps". *IEEE transactions on visualization and computer graphics*, Vol. 21, No. 11, pp. 1309–1318, 2015.

[Ash 08] J. N. Ash and R. L. Moses. "On optimal anchor node placement in sensor localization by optimization of subspace principal angles". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2289–2292, IEEE, 2008.

[Bacc 12] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves. "Radio link quality estimation in wireless sensor networks: a survey". *ACM Transactions on Sensor Networks (TOSN)*, Vol. 8, No. 4, p. 34, 2012.

[Bash 13] M. R. Basheer and S. Jagannathan. "Localization of RFID tags using stochastic tunneling". *IEEE Transactions on Mobile Computing*, Vol. 12, No. 6, pp. 1225–1235, 2013.

[Belk 13] F. Belkhouche. "Modeling and Calculating the Collision Risk for Air Vehicles". *IEEE transactions on vehicular technology*, Vol. 62, No. 5, pp. 2031–2041, 2013.

[Blom 06] H. A. Blom, J. Krystul, and G. B. Bakker. "A particle system for safety verification of free flight in air traffic". In: *45th IEEE Conference on Decision and Control*, pp. 1574–1579, IEEE, 2006.

[Blum 05] J. Blumenthal, F. Reichenbach, and D. Timmermann. "Precise positioning with a low complexity algorithm in ad hoc wireless sensor networks". *Praxis Der Informationsverarbeitung Und Kommunikation*, Vol. 28, No. 2, pp. 80–85, 2005.

[Blum 07] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. "Weighted centroid localization in zigbee-based sensor networks". In: *IEEE International Symposium on Intelligent Signal Processing*, pp. 1–6, IEEE, 2007.

[Bouk 07] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro. "Localization systems for wireless sensor networks". *IEEE wireless Communications*, Vol. 14, No. 6, 2007.

[Bulu 00] N. Bulusu, J. Heidemann, and D. Estrin. “GPS-less low-cost outdoor localization for very small devices”. *IEEE personal communications*, Vol. 7, No. 5, pp. 28–34, 2000.

[Carl 13] R. Carloni, V. Lippiello, M. D’auria, M. Fumagalli, A. Y. Mersha, S. Stramigioli, and B. Siciliano. “Robot vision: Obstacle-avoidance techniques for unmanned aerial vehicles”. *IEEE Robotics & Automation Magazine*, Vol. 20, No. 4, pp. 22–31, 2013.

[Carp 97] B. Carpenter and J. K. Kuchar. “Probability-based collision alerting logic for closely-spaced parallel approach”. In: *Proceedings of the AIAA 35th Aerospace Sciences Meeting and Exhibit*, pp. 97–0222, Reno, 1997.

[Chai 07] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. “Impact of human mobility on opportunistic forwarding algorithms”. *IEEE Transactions on Mobile Computing*, Vol. 6, No. 6, pp. 606–620, 2007.

[Chau 14] V. K. Chaurasiya, N. Jain, and G. C. Nandi. “A novel distance estimation approach for 3D localization in wireless sensor network using multi dimensional scaling”. *Information Fusion*, Vol. 15, pp. 5–18, 2014.

[Chau 16] S. Chaudhari and D. Cabric. “Cyclic Weighted Centroid Algorithm for Transmitter Localization in the Presence of Interference”. *IEEE Transactions on Cognitive Communications and Networking*, Vol. 2, No. 2, pp. 162–177, 2016.

[Chen 10] H. Chen, Q. Shi, R. Tan, H. V. Poor, and K. Sezaki. “Mobile element assisted cooperative localization for wireless sensor networks with obstacles”. *IEEE transactions on wireless communications*, Vol. 9, No. 3, pp. 956–963, 2010.

[Chen 11] L. Cheng, C.-D. Wu, and Y.-Z. Zhang. “Indoor robot localization based on wireless sensor networks”. *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 3, pp. 1099–1104, 2011.

[Chen 12a] H. Chen, B. Liu, P. Huang, J. Liang, and Y. Gu. “Mobility-assisted node localization based on TOA measurements without time synchronization in wireless sensor networks”. *Mobile Networks and Applications*, Vol. 17, No. 1, pp. 90–99, 2012.

[Chen 12b] H. Chen, G. Wang, Z. Wang, H.-C. So, and H. V. Poor. “Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks”. *IEEE Transactions on Wireless Communications*, Vol. 11, No. 1, pp. 108–116, 2012.

[Chen 13] H. Chenji and R. Stoleru. “Toward accurate mobile sensor network localization in noisy environments”. *IEEE transactions on Mobile Computing*, Vol. 12, No. 6, pp. 1094–1106, 2013.

[Choe 14] T.-S. Choe, J.-B. Park, S.-H. Joo, and Y.-W. Park. “1D virtual force field algorithm for reflexive local path planning of mobile robots”. *Electronics Letters*, Vol. 50, No. 20, pp. 1429–1430, 2014.

[Choi 11] B.-S. Choi, J.-W. Lee, J.-J. Lee, and K.-T. Park. “A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion”. *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 6, pp. 2226–2235, 2011.

[Chry 11] J. P. Chryssanthacopoulos and M. J. Kochenderfer. “Accounting for state uncertainty in collision avoidance”. *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, pp. 951–960, 2011.

[Cond 12] R. Conde, D. Alejo, J. A. Cobano, A. Viguria, and A. Ollero. “Conflict detection and resolution method for cooperating unmanned aerial vehicles”. *Journal of Intelligent & Robotic Systems*, Vol. 65, No. 1, pp. 495–505, 2012.

[Corr 16] A. Correa, M. B. Llado, A. Morell, and J. L. Vicario. “Indoor pedestrian tracking by on-body multiple receivers”. *IEEE Sensors Journal*, Vol. 16, No. 8, pp. 2545–2553, 2016.

[Cui 12] H. Cui and Y. Wang. “Four-mobile-beacon assisted localization in three-dimensional wireless sensor networks”. *Computers & Electrical Engineering*, Vol. 38, No. 3, pp. 652–661, 2012.

[Datt 06] S. Datta, C. Klinowski, M. Rudafshani, and S. Khaleque. “Distributed localization in static and mobile sensor networks”. In: *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 69–76, IEEE, 2006.

[DiGi 14] E. DiGiampaolo and F. Martinelli. “Mobile robot localization using the phase of passive UHF RFID signals”. *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 1, pp. 365–376, 2014.

[Dorf 15] Y. Dorfan and S. Gannot. “Tree-based recursive expectation-maximization algorithm for localization of acoustic sources”. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 23, No. 10, pp. 1692–1703, 2015.

[Dowe 05] G. Dowek, C. Munoz, and V. A. Carreno. “Provably safe coordinated strategy for distributed conflict resolution”. In: *Proceedings of the AIAA guidance navigation, and control conference and exhibit*, 2005.

[Dowe 07] G. Dowek and C. Munoz. “Conflict detection and resolution for 1, 2,..., N aircraft”. In: *Proceedings of the 7th AIAA Aviation, Technology, Integration, and Operations Conference, Belfast, Northern Ireland*, 2007.

[Du T 11] N. E. Du Toit and J. W. Burdick. “Probabilistic collision checking with chance constraints”. *IEEE Transactions on Robotics*, Vol. 27, No. 4, pp. 809–815, 2011.

[Duon 96] V. Duong, E. Hoffman, L. Flochic, J. Nicolaon, and A. Bossu. “Extended Rules-Of-The-Air To Apply To The Resolution Of Encounters In Autonomous Airborne Separation”. *Eurocontrol Experimental Center Rep*, 1996.

[Fang 15] S.-H. Fang, Y.-T. Hsu, Y. Shiao, and F.-Y. Sung. “An Enhanced Device Localization Approach Using Mutual Signal Strength in Cellular Networks”. *IEEE Internet of Things Journal*, Vol. 2, No. 6, pp. 596–603, 2015.

[Feng 12] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan. “Received-signal-strength-based indoor positioning using compressive sensing”. *IEEE Transactions on Mobile Computing*, Vol. 11, No. 12, pp. 1983–1993, 2012.

[Filo 10] V. Filonenko, C. Cullen, and J. Carswell. “Investigating ultrasonic positioning on mobile phones”. In: *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, 2010.

[Fu 11] Y.-J. Fu, T.-H. Lee, L.-h. Chang, and T.-P. Wang. “A single mobile anchor localization scheme for wireless sensor networks”. In: *IEEE 13th International Conference on High Performance Computing and Communications (HPCC)*, pp. 946–950, IEEE, 2011.

[Gold 05] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[Gran 01] G. Granger, N. Durand, J.-M. Alliot, *et al.* “Token Allocation Strategy for Free-Flight Conflict Solving.”. In: *IAAI*, pp. 59–64, 2001.

[Grib 14] J. Gribben and A. Boukerche. “Location error estimation in wireless ad hoc networks”. *Ad Hoc Networks*, Vol. 13, pp. 504–515, 2014.

[Groe 05] R. Groenevelt, P. Nain, and G. Koole. “The message delay in mobile ad hoc networks”. *Performance Evaluation*, Vol. 62, No. 1, pp. 210–228, 2005.

[Gu 15] Y. Gu and F. Ren. “Energy-efficient indoor localization of smart hand-held devices using Bluetooth”. *IEEE Access*, Vol. 3, pp. 1450–1461, 2015.

[Guo 10] Z. Guo, Y. Guo, F. Hong, Z. Jin, Y. He, Y. Feng, and Y. Liu. “Perpendicular intersection: locating wireless sensors with mobile beacon”. *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 7, pp. 3501–3509, 2010.

[Hadd 16] D. B. Haddad, W. A. Martins, M. d. V. da Costa, L. W. Biscainho, L. O. Nunes, and B. Lee. “Robust Acoustic Self-Localization of Mobile Devices”. *IEEE Transactions on Mobile Computing*, Vol. 15, No. 4, pp. 982–995, 2016.

[Hai 11] C. Hai-qing, W. Hua, and W. Hua-kui. “An improved centroid localization algorithm based on weighted average in WSN”. In: *3rd International Conference on Electronics Computer Technology (ICECT)*, pp. 258–262, IEEE, 2011.

[Halp 11] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. “Predictable 802.11 packet delivery from wireless channel measurements”. *ACM SIGCOMM Computer Communication Review*, Vol. 41, No. 4, pp. 159–170, 2011.

[Han 14] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, P. Zhang, W. Xi, and Z. Jiang. “Twins: Device-free object tracking using passive tags”. In: *IEEE Conference on Computer Communications*, pp. 469–476, IEEE, 2014.

[Hard 13] J. Hardy and M. Campbell. “Contingency planning over probabilistic obstacle predictions for autonomous road vehicles”. *IEEE Transactions on Robotics*, Vol. 29, No. 4, pp. 913–929, 2013.

[He 03] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. “Range-free localization schemes for large scale sensor networks”. In: *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95, ACM, 2003.

[He 16] S. He and S.-H. G. Chan. “Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons”. *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 1, pp. 466–490, 2016.

[Hoff 08] G. M. Hoffmann and C. J. Tomlin. “Decentralized cooperative collision avoidance for acceleration constrained vehicles”. In: *47th IEEE Conference on Decision and Control*, pp. 4357–4363, IEEE, 2008.

[Hu 04] L. Hu and D. Evans. “Localization for mobile sensor networks”. In: *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 45–57, ACM, 2004.

[Huan 15] J.-F. Huang, G.-Y. Chang, and G.-H. Chen. “A historical-beacon-aided localization algorithm for mobile sensor networks”. *IEEE Transactions on Mobile Computing*, Vol. 14, No. 6, pp. 1109–1122, 2015.

[Hui 05] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. “Pocket switched networks and human mobility in conference environments”. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 244–251, ACM, 2005.

[Hwan 08] I. Hwang and C. E. Seah. “Intent-based probabilistic conflict detection for the next generation air transportation system”. *Proceedings of the IEEE*, Vol. 96, No. 12, pp. 2040–2059, 2008.

[Hyyt 05] E. Hyytiä and J. Virtamo. “Random waypoint model in n-dimensional space”. *Operations Research Letters*, Vol. 33, No. 6, pp. 567–571, 2005.

[Igle 12] H. J. P. Iglesias, V. Barral, and C. J. Escudero. “Indoor person localization system through RSSI Bluetooth fingerprinting”. In: *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 40–43, IEEE, 2012.

[Irfa 10] N. Irfan, M. Bolic, M. C. Yagoub, and V. Narasimhan. “Neural-based approach for localization of sensors in indoor environment”. *Telecommunication Systems*, Vol. 44, No. 1, pp. 149–158, 2010.

[Isaa 97] D. R. Isaacson and H. Erzberger. “Design of a conflict detection algorithm for the Center/TRACON automation system”. In: *16th DASC. AIAA/IEEE Digital Avionics Systems Conference*, pp. 9–3, IEEE, 1997.

[Jaco 15] M. Jacobsson and C. Rohner. “Estimating Packet Delivery Ratio for Arbitrary Packet Sizes Over Wireless Links”. *IEEE Communications Letters*, Vol. 19, No. 4, pp. 609–612, 2015.

[Jung 11] S.-Y. Jung, S. Hann, and C.-S. Park. “TDOA-based optical wireless indoor localization using LED ceiling lamps”. *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 4, 2011.

[Kara 10] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic. “Power law and exponential decay of intercontact times between mobile devices”. *IEEE Transactions on Mobile Computing*, Vol. 9, No. 10, pp. 1377–1390, 2010.

[Kase 11] V. Kaseva, T. D. Hämäläinen, and M. Hännikäinen. “Range-free algorithm for energy-efficient indoor localization in wireless sensor networks”. In: *Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pp. 1–8, IEEE, 2011.

[Kim 10] E. Kim, S. Lee, C. Kim, and K. Kim. “Mobile beacon-based 3D-localization with multidimensional scaling in large sensor networks”. *IEEE Communications Letters*, Vol. 14, No. 7, pp. 647–649, 2010.

[King 06] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. “Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses”. In: *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, pp. 34–40, ACM, 2006.

[Kuch 00] J. K. Kuchar and L. C. Yang. “A review of conflict detection and resolution modeling methods”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, pp. 179–189, 2000.

[Kuwa 14] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. “Safe maritime autonomous navigation with colregs, using velocity obstacles”. *IEEE Journal of Oceanic Engineering*, Vol. 39, No. 1, pp. 110–119, 2014.

[Lam 09] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. van Paassen. “Artificial Force Field for Haptic Feedback in UAV Teleoperation”. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 39, No. 6, pp. 1316–1330, Nov 2009.

[Lamb 08] A. Lambert, D. Gruyer, and G. S. Pierre. “A fast monte carlo algorithm for collision probability estimation”. In: *10th International Conference on Control, Automation, Robotics and Vision*, pp. 406–411, IEEE, 2008.

[Lass 06] P. Lassila, J. Virtamo, *et al.* “Spatial node distribution of the random waypoint mobility model with applications”. *IEEE Transactions on mobile computing*, No. 6, pp. 680–694, 2006.

[Leve 11] S. Leven, J.-C. Zufferey, and D. Floreano. “Dealing with midair collisions in dense collective aerial systems”. *Journal of Field Robotics*, Vol. 28, No. 3, pp. 405–423, 2011.

[Lim 10] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. “Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure”. *Wireless Networks*, Vol. 16, No. 2, pp. 405–420, 2010.

[Lin 15] Z. Lin, M. Fu, and Y. Diao. “Distributed self localization for relative position sensing networks in 2D space”. *IEEE Transactions on Signal Processing*, Vol. 63, No. 14, pp. 3751–3761, 2015.

[Liu 10] W. Liu and I. Hwang. “Probabilistic 4D trajectory prediction and conflict detection for air traffic control”. In: *49th IEEE Conference on Decision and Control (CDC)*, pp. 1183–1188, IEEE, 2010.

[Liu 11] W. Liu and I. Hwang. “Probabilistic trajectory prediction and conflict detection for air traffic control”. *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, pp. 1779–1789, 2011.

[Luo 13] C. Luo, S. I. McClean, G. Parr, L. Teacy, and R. De Nardi. “UAV position estimation and collision avoidance using the extended Kalman filter”. *IEEE Transactions on Vehicular Technology*, Vol. 62, No. 6, pp. 2749–2762, 2013.

[Manj 13] D. Manjarres, J. Del Ser, S. Gil-Lopez, M. Vecchio, I. Landa-Torres, and R. Lopez-Valcarce. “A novel heuristic approach for distance-and connectivity-based multihop node localization in wireless sensor networks”. *Soft Computing*, Vol. 17, No. 1, pp. 17–28, 2013.

[Meng 14] W. Meng, D. Zhang, Y. Wang, and C. Li. “An extended centroid localization algorithm based on error correction in WSN”. In: *IEEE Global Communications Conference*, pp. 442–447, IEEE, 2014.

[Meye 16] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch. “Distributed localization and tracking of mobile networks including noncooperative objects”. *IEEE Transactions on Signal and Information Processing over Networks*, Vol. 2, No. 1, pp. 57–71, 2016.

[Mond 01] S. Mondoloni and S. Conway. “An airborne conflict resolution approach using a genetic algorithm”. In: *AIAA Guidance, Navigation, and Control Conf., Montreal*, 2001.

[Muno 10] C. A. Munoz and A. J. Narkawicz. “Time of Closest Approach in Three-Dimensional Airspace”. Tech. Rep., National Aeronautics and Space Administration, Langley Research Center, 2010.

[Muno 13] C. Munoz, A. Narkawicz, and J. Chamberlain. “A TCAS-II resolution advisory detection algorithm”. In: *Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit*, 2013.

[Nain 05] P. Nain, D. Towsley, B. Liu, and Z. Liu. “Properties of random direction models”. In: *Proceedings of IEEE INFOCOM*, pp. 1897–1907, IEEE, 2005.

[Nicu 01] D. Niculescu and B. Nath. “Ad hoc positioning system (APS)”. In: *IEEE Global Telecommunications Conference*, pp. 2926–2931, IEEE, 2001.

[Ning 12] J. Ning, S. Singh, K. Pelechrinis, B. Liu, S. V. Krishnamurthy, and R. Govindan. “Forensic analysis of packet losses in wireless networks”. In: *20th IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10, IEEE, 2012.

[Oguz 14] P. Oğuz-Ekim, J. P. Gomes, J. Xavier, M. Stošić, and P. Oliveira. “An Angular Approach for Range-Based Approximate Maximum Likelihood Source Localization Through Convex Relaxation”. *IEEE Transactions on Wireless Communications*, Vol. 13, No. 7, pp. 3951–3964, 2014.

[Oksa 14] I. Oksar. “A Bluetooth signal strength based indoor localization method”. In: *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 251–254, IEEE, 2014.

[Oliv 14] L. Oliveira, H. Li, L. Almeida, and T. E. Abrudan. “RSSI-based relative localisation for mobile robots”. *Ad Hoc Networks*, Vol. 13, pp. 321–335, 2014.

[Ou 08a] C.-H. Ou. “Range-free node localization for mobile wireless sensor networks”. In: *3rd International Symposium on Wireless Pervasive Computing, ISWPC 2008*, pp. 535–539, IEEE, 2008.

[Ou 08b] C.-H. Ou and K.-F. Ssu. “Sensor position determination with flying anchors in three-dimensional wireless sensor networks”. *IEEE Transactions on Mobile Computing*, Vol. 7, No. 9, pp. 1084–1097, 2008.

[Paie 97] R. A. Paielli and H. Erzberger. “Conflict Probability for Free Flight”. *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, pp. 588–596, 1997.

[Pati 12] S. Patil, J. Van den Berg, and R. Alterovitz. "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3238–3244, IEEE, 2012.

[Piva 11] P. Pivato, L. Palopoli, and D. Petri. "Accuracy of RSS-based centroid localization algorithms in an indoor environment". *IEEE Transactions on Instrumentation and Measurement*, Vol. 60, No. 10, pp. 3451–3460, 2011.

[Priy 00] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. "The cricket location-support system". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 32–43, ACM, 2000.

[Qian 14] L. Qian, D. Qingzhi, L. Hua, and S. Yubin. "A new RSSI-based centroid localization algorithm using virtual reference tags". *Application of Electronic Technique*, Vol. 3, p. 040, 2014.

[Ragh 10] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran. "Accurate mobile robot localization in indoor environments using bluetooth". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4391–4396, IEEE, 2010.

[Rapp 96] T. S. Rappaport *et al. Wireless communications: principles and practice*. Vol. 2, Prentice Hall PTR New Jersey, 1996.

[Rebo 07] J. J. Rebollo, A. Ollero, and I. Maza. "Collision avoidance among multiple aerial robots and other non-cooperative aircraft based on velocity planning". In: *Proceedings of ROBOTICA 2007 Conference, Paderne, Portugal*, Citeseer, 2007.

[Reyn 13] E. Arias-de Reyna and P. M. Djurić. "Indoor localization with range-based measurements and little prior information". *IEEE Sensors Journal*, Vol. 13, No. 5, pp. 1979–1987, 2013.

[Reza 11] J. Rezazadeh, M. Moradi, and A. S. Ismail. "Efficient localization via middle-node cooperation in wireless sensor networks". In: *International Conference on Electrical, Control and Computer Engineering (INECCE)*, pp. 410–415, IEEE, 2011.

[Reza 14] H. Rezaee and F. Abdollahi. "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots". *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 1, pp. 347–354, 2014.

[Rhee 09] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong. "CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23)". Downloaded from `http://crawdad.org/ncsu/mobilitymodels/20090723`, July 2009.

[Roy 10] R. R. Roy. *Handbook of mobile ad hoc networks for mobility models*. Springer Science & Business Media, 2010.

[Ruda 07] M. Rudafshani and S. Datta. "Localization in wireless sensor networks". In: *6th International Symposium on Information Processing in Sensor Networks*, pp. 51–60, IEEE, 2007.

[Saab 11] S. S. Saab and Z. S. Nakad. "A standalone RFID indoor positioning system using passive tags". *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 5, pp. 1961–1970, 2011.

[Saad 07] C. Saad, A. Benslimane, J. König, J. Turbert, *et al.* "At-free: A preliminary method for localization techniques in sensor networks". In: *Proceedings of the 7th IEEE International Conference on New Technologies of Distributed Systems*, 2007.

[Saad 08] C. Saad, A. Benslimane, and J. König. "AT-DIST: A Distributed Method for Localization with high accuracy in Sensor Networks". *International journal Studia Informatica Universalis, Special Issue on'Wireless Ad Hoc and Sensor Networks'*, Vol. 6, No. 1, 2008.

[Saha 13] L. R. Sahawneh, R. W. Beard, S. Avadhanam, and H. Bai. "Chain-based Collision Avoidance for UAS Sense-and-Avoid Systems". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 19–22, 2013.

[Saha 14] L. R. Sahawneh and R. W. Beard. "A probabilistic framework for unmanned aircraft systems collision detection and risk estimation". In: *53rd IEEE Conference on Decision and Control*, pp. 242–247, IEEE, 2014.

[Saha 15] L. R. Sahawneh, J. Mackie, J. Spencer, R. W. Beard, and K. F. Warnick. "Airborne radar-based collision detection and risk estimation for small unmanned aircraft systems". *Journal of Aerospace Information Systems*, Vol. 12, No. 12, pp. 756–766, 2015.

[Sahi 15] A. Şahin, Y. S. Eroğlu, I. Güvenç, N. Pala, and M. Yüksel. "Hybrid 3-D Localization for Visible Light Communication Systems". *Journal of Lightwave Technology*, Vol. 33, No. 22, pp. 4589–4599, 2015.

[Sahu 13] P. K. Sahu, E. H.-K. Wu, and J. Sahoo. "DuRT: dual RSSI trend based localization for wireless sensor networks". *IEEE Sensors Journal*, Vol. 13, No. 8, pp. 3115–3123, 2013.

[Sala 13] S. Salari, S. Shahbazpanahi, and K. Ozdemir. "Mobility-aided wireless sensor network localization via semidefinite programming". *IEEE Transactions on Wireless Communications*, Vol. 12, No. 12, pp. 5966–5978, 2013.

[Sath 11] T. Sathyan, D. Humphrey, and M. Hedley. "WASP: A system and algorithms for accurate radio localization using low-cost hardware". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 41, No. 2, pp. 211–222, 2011.

[Savi 16] V. Savic, H. Wymeersch, and E. G. Larsson. "Target tracking in confined environments with uncertain sensor positions". *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 2, pp. 870–882, 2016.

[Sche 13] M. Scherhäufl, M. Pichler, E. Schimbäck, D. J. Müller, A. Ziroff, and A. Stelzer. "Indoor localization of passive UHF RFID tags based on phase-of-arrival evaluation". *IEEE Transactions on Microwave Theory and Techniques*, Vol. 61, No. 12, pp. 4724–4729, 2013.

[Sedg 13] R. Sedgewick and P. Flajolet. *An introduction to the analysis of algorithms*. Addison-Wesley, 2013.

[Seet 12] B.-C. Seet, Q. Zhang, C. H. Foh, and A. C. Fong. "Hybrid RF mapping and Kalman filtered spring relaxation for sensor network localization". *IEEE Sensors Journal*, Vol. 12, No. 5, pp. 1427–1435, 2012.

[Seke 15] Y. A. Şekerciogğlu, J. Violi, L. Priestnall, and J. Armstrong. "Accurate node localization with directional pulsed infrared light for indoor ad hoc network applications". In: *22nd International Conference on Telecommunications (ICT)*, pp. 384–390, IEEE, 2015.

[Shao 14] H.-J. Shao, X.-P. Zhang, and Z. Wang. "Efficient closed-form algorithms for AOA based self-localization of sensor nodes using auxiliary variables". *IEEE Transactions on Signal Processing*, Vol. 62, No. 10, pp. 2580–2594, 2014.

[Shen 12] J. Shen, A. F. Molisch, and J. Salmi. "Accurate passive location estimation using TOA measurements". *IEEE Transactions on Wireless Communications*, Vol. 11, No. 6, pp. 2182–2192, 2012.

[Sheu 10] J.-P. Sheu, W.-K. Hu, and J.-C. Lin. "Distributed localization scheme for mobile sensor networks". *IEEE Transactions on Mobile Computing*, Vol. 9, No. 4, pp. 516–526, 2010.

[Shyu 08] H.-J. Shyur. "A quantitative model for aviation safety risk assessment". *Computers & Industrial Engineering*, Vol. 54, No. 1, pp. 34–44, 2008.

[Simo 14] A. Simonetto and G. Leus. "Distributed Maximum Likelihood Sensor Network Localization.". *IEEE Trans. Signal Processing*, Vol. 62, No. 6, pp. 1424–1437, 2014.

[Sisl 06] D. Sislak, M. Rehak, M. Pechoucek, D. s. Pavlicek, and M. Uller. "Negotiation-based approach to unmanned aerial vehicles". In: *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pp. 279–284, IEEE, 2006.

[Sisl 07] D. Sislak, P. Volf, A. Komenda, J. Samek, and M. Pechoucek. "Agent-based multi-layer collision avoidance to unmanned aerial vehicles". In: *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 365–370, IEEE, 2007.

[Snap 10] J. Snape, J. Van den Berg, S. J. Guy, and D. Manocha. "Smooth and collision-free navigation for multiple robots under differential-drive constraints". In: *EEE/RSJ International Conference on Intelligent Robots and Systems*, Citeseer, 2010.

[Song 13] S. Song, C. Hu, B. Li, X. Li, and M.-H. Meng. "An electromagnetic localization and orientation method based on rotating magnetic dipole". *IEEE Transactions on Magnetics*, Vol. 49, No. 3, pp. 1274–1277, 2013.

[Stro 09] S. H. Stroeve, H. A. Blom, and G. B. Bakker. "Systemic accident risk assessment in air traffic by Monte Carlo simulation". *Safety science*, Vol. 47, No. 2, pp. 238–249, 2009.

[Suh 16] J. Suh, S. You, S. Choi, and S. Oh. "Vision-based coordinated localization for mobile sensor networks". *IEEE transactions on automation science and engineering*, Vol. 13, No. 2, pp. 611–620, 2016.

[Tomi 15] S. Tomic, M. Beko, and R. Dinis. "RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes". *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 5, pp. 2037–2050, 2015.

[Tomi 16] S. Tomic, M. Beko, and R. Dinis. “Distributed RSS-AoA based localization with unknown transmit powers”. *IEEE Wireless Communications Letters*, Vol. 5, No. 4, pp. 392–395, 2016.

[Toml 98] C. Tomlin, G. J. Pappas, and S. Sastry. “Conflict resolution for air traffic management: A study in multiagent hybrid systems”. *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, pp. 509–521, 1998.

[Varg 01] A. Varga *et al.* “The OMNeT++ discrete event simulation system”. In: *Proceedings of the European simulation multiconference (ESM 2001)*, p. 65, sn, 2001.

[Varg 08] A. Varga and R. Hornig. “An overview of the OMNeT++ simulation environment”. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, p. 60, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[Wang 10] J. Wang, R. K. Ghosh, and S. K. Das. “A survey on sensor localization”. *Journal of Control Theory and Applications*, Vol. 8, No. 1, pp. 2–11, 2010.

[Wang 11a] G. Wang and K. Yang. “A new approach to sensor node localization using RSS measurements in wireless sensor networks”. *IEEE transactions on wireless communications*, Vol. 10, No. 5, pp. 1389–1395, 2011.

[Wang 11b] J. Wang, P. Urriza, Y. Han, and D. Cabric. “Weighted centroid localization algorithm: theoretical analysis and distributed implementation”. *IEEE Transactions on Wireless Communications*, Vol. 10, No. 10, pp. 3403–3413, 2011.

[Wang 15] Y. Wang and K. Ho. “An Asymptotically Efficient Estimator in Closed-Form for 3-D AOA Localization Using a Sensor Network”. *IEEE Transactions on Wireless Communications*, Vol. 14, No. 12, pp. 6524–6535, 2015.

[Wen 10] C.-Y. Wen and F.-K. Chan. “Adaptive AOA-aided TOA self-positioning for mobile wireless sensor networks”. *Sensors*, Vol. 10, No. 11, pp. 9742–9770, 2010.

[Wen 12] S. Wen, W. Zheng, J. Zhu, X. Li, and S. Chen. “Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation”. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 42, No. 4, pp. 603–608, 2012.

[Wu 13] C. Wu, Z. Yang, Y. Liu, and W. Xi. "WILL: Wireless indoor localization without site survey". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 4, pp. 839–848, 2013.

[Xiao 10a] B. Xiao, L. Chen, Q. Xiao, and M. Li. "Reliable anchor-based sensor localization in irregular areas". *IEEE Transactions on Mobile Computing*, Vol. 9, No. 1, pp. 60–72, 2010.

[Xiao 10b] Q. Xiao, B. Xiao, J. Cao, and J. Wang. "Multihop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme". *IEEE Transactions on Mobile Computing*, Vol. 9, No. 11, pp. 1592–1607, 2010.

[Xu 11] E. Xu, Z. Ding, and S. Dasgupta. "Source localization in wireless sensor networks from signal time-of-arrival measurements". *IEEE Transactions on Signal Processing*, Vol. 59, No. 6, pp. 2887–2897, 2011.

[Yagh 14] F. Yaghoubi, A.-A. Abbasfar, and B. Maham. "Energy-efficient RSSI-based localization for wireless sensor networks". *IEEE Communications Letters*, Vol. 18, No. 6, pp. 973–976, 2014.

[Yang 02] L. C. Yang and J. K. Kuchar. "Performance metric alerting: A new design approach for complex alerting problems". *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 32, No. 1, pp. 123–134, 2002.

[Yang 10] X. Yang, Q. Kong, and X. Dai. "An Improved Weighted Centroid Location Algorithm". *Journal of Xi'an Jiaotong University*, Vol. 8, p. 002, 2010.

[Yang 13] X. Yang, L. M. Alvarez, and T. Bruggemann. "A 3D collision avoidance strategy for UAVs in a non-cooperative environment". *Journal of Intelligent & Robotic Systems*, Vol. 70, No. 1-4, pp. 315–327, 2013.

[Yang 15] C. Yang and H.-R. Shao. "WiFi-based indoor positioning". *IEEE Communications Magazine*, Vol. 53, No. 3, pp. 150–157, 2015.

[Yang 97] L. C. Yang and J. K. Kuchar. "Prototype conflict alerting system for free flight". *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, pp. 768–773, 1997.

[Yepe 07] J. L. Yepes, I. Hwang, and M. Rotea. "New algorithms for aircraft intent inference and trajectory prediction". *Journal of guidance, control, and dynamics*, Vol. 30, No. 2, pp. 370–382, 2007.

[Zaid 16] S. Zaidi, A. El Assaf, S. Affes, and N. Kandil. "Accurate Range-Free Localization in Multi-Hop Wireless Sensor Networks". *IEEE Transactions on Communications*, Vol. 64, No. 9, pp. 3886–3900, 2016.

[Zhan 10a] B. Zhang and F. Yu. "LSWD: Localization scheme for wireless sensor networks using directional antenna". *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 4, 2010.

[Zhan 10b] S. Zhang, J. Cao, C. Li-Jun, and D. Chen. "Accurate and energy-efficient range-free localization for mobile sensor networks". *IEEE Transactions on Mobile Computing*, Vol. 9, No. 6, pp. 897–910, 2010.

[Zhan 13] D. Zhang and H. Yin. "Study on Centroid Localization Algorithm Based on the Cellular Distribution of Anchor Nodes". In: *Joint International Conference on Pervasive Computing and the Networked World*, pp. 783–789, Springer, 2013.

[Zhao 12] Y. Zhao, N. Patwari, P. Agrawal, and M. Rabbat. "Directed by directionality: Benefiting from the gain pattern of active RFID badges". *IEEE Transactions on Mobile Computing*, Vol. 11, No. 5, pp. 865–877, 2012.

[Zhao 13] J. Zhao, W. Xi, Y. He, Y. Liu, X.-Y. Li, L. Mo, and Z. Yang. "Localization of wireless sensor networks in the wild: Pursuit of ranging quality". *IEEE/ACM Transactions on Networking (TON)*, Vol. 21, No. 1, pp. 311–323, 2013.

[Zhon 11] Z. Zhong and T. He. "RSD: A metric for achieving range-free localization beyond connectivity". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 11, pp. 1943–1951, 2011.

[Zhou 11a] J. Zhou, H. Zhang, and L. Mo. "Two-dimension localization of passive RFID tags using AOA estimation". In: *IEEE International Instrumentation and Measurement Technology Conference*, pp. 1–5, IEEE, 2011.

[Zhou 11b] Y. Zhou, C. L. Law, Y. L. Guan, and F. Chin. "Indoor elliptical localization based on asynchronous UWB range measurement". *IEEE Transactions on Instrumentation and Measurement*, Vol. 60, No. 1, pp. 248–257, 2011.

[Zhu 10] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni. "Recognizing exponential inter-contact time in VANETs". In: *Proceedings of IEEE INFOCOM*, pp. 1–5, IEEE, 2010.

[Zou 14] D. Y. Zou, T. F. Han, D. L. Zheng, and H. Lv. “Centroid Localization Algorithm Based on the Anchor Nodes Cellular Distribution”. In: *Applied Mechanics and Materials*, pp. 579–582, Trans Tech Publ, 2014.