Université Lille 1 – Sciences et Technologies

UFR IEEA **École Doctorale SPI 072**

THÈSE

Présentée pour obtenir le titre de :

Docteur en Sciences de l'Université Lille 1

Spécialité : Informatique

par

Karen Samara MIRANDA CAMPOS

Équipe d'accueil : FUN – Inria

Algorithmes d'auto-déploiement adaptatifs pour des réseaux de substitution mobiles sans fil

Soutenue publiquement le 10 décembre 2013 à 14h30 devant le jury composé de :

Présidente :	**Isabelle GUÉRIN LASSOUS** Université de Lyon I, France
Rapporteurs :	**Anne FLADENMULLER** Université Pierre et Marie Curie, France **Juan Carlos CANO ESCRIBÁ** Universitat Politèctica de València, Espagne
Examinateur :	**Thierry DELOT** Université de Valenciennes, France
Directeur de thèse :	**David SIMPLOT-RYL** Université Lille 1, France
Co-encadrant de thèse :	**Tahiry RAZAFINDRALAMBO** Inria Lille – Nord Europe, France

Inria Lille - Nord Europe

Numéro d'ordre: 41206

Algorithmes d'auto-déploiement adaptatifs pour des réseaux de substitution mobiles sans fil

Titre de la thèse en anglais :

Adaptive self-deployment algorithms for mobile wireless substitution networks

Karen Samara MIRANDA CAMPOS

10 décembre 2013

À Víctor, à ma mère
et en souvenir de mon père

Remerciements

Je remercie tout d'abord Isabelle Guérin-Lassous de me faire l'honneur de présider mon jury. Je remercie également Anne Fladenmuller et Juan Carlos Cano de s'être intéressés à ce travail et d'avoir bien voulu accepter la charge de rapporteur pour cette thèse. Un grand merci à Thierry Delot d'avoir accepté de participer à ce jury, mais aussi et surtout merci pour sa gentillesse et sa bonne humeur en tant que membre de l'équipe FUN.

Je tiens à exprimer ma gratitude envers Tahiry Razafindralambo et David Simplot-Ryl qui m'ont guidée durant la préparation de cette thèse. Merci de m'avoir proposé ce sujet de thèse et pour l'autonomie qu'ils m'ont donnée.

J'ai eu la chance de faire partie de l'équipe FUN, très accueillante et multiculturelle. Mes remerciements s'adressent à tous mes collègues : les Espagnols, les Français, les Grecs, les Italiens, les Malgaches, les Serbes, les Sud-Africains, les Tunisiens avec qui j'ai pris plaisir à travailler. Merci particulièrement à Enrico Natalizio, Tanel Razafimandimby, Dimitris Zorbas, Milan Erdelj et Kalypso Magklara. Je suis très reconnaissante à Nathalie Mitton, responsable de l'équipe, pour sa gentillesse et son professionnalisme. Merci de m'avoir soutenue au moment le plus délicat de ma thèse.

Je tiens vivement à remercier Antonella Molinaro et Antonio Iera de m'avoir accueillie au sein de leur laboratoire à Reggio de Calabre en Italie. Je voudrais également remercier les membres du ARTS Laboratory. Je pense particulièrement à Claudia, Marica, Sara, Maria, Chiara, Leonardo, Giussepe et Nicola pour leur gentillesse.

Au long de ces années, j'ai rencontré des gens merveilleux devenus mes amis. Je remercie chaleureusement Jenya, Kate, Raquel, Anke, Ewa, Di, Wei, Fabio et Philipp.

Mes remerciements vont bien évidemment à ma famille. À tous mes cousins, cousines, oncles, tantes, nièces et neveux. Spécialement, je remercie ma mère, mes sœurs et mes beaux-frères de m'avoir épaulée pendant ces années. Merci à ma sœur Claudia d'avoir donné naissance aux triplés. Même si ce n'est pas le but, ils sont toujours une source de bonheur pour moi.

Enfin, j'aimerais remercier Víctor pour son soutien, sa patience, ses encouragements et l'ensemble de ses conseils. Je le remercie de tout mon cœur pour le temps qu'il m'a consacré.

Abstract

In case of a disaster, the communication infrastructure may be partially or totally destroyed, or scarce because of the high data traffic. Even so, it is necessary to provide connectivity between the rescue teams and the command center. Therefore, temporary communication solutions are crucial until the infrastructure is restored. In this thesis, we focus on the deployment of a communication solution in such a context called substitution networks. Thus, we propose a self-deployment algorithm to allow mobile routers that compose a substitution network to be spread out to cover the target area. Our algorithm monitors the network conditions to decide whether the router should move or not, adjusting its position based on one-hop information by means of active measurement, i.e., probe packets. Such probe packets allow the algorithm to monitor the channel and its eventual changes over time. If the probe transmission rate is high enough, the insights obtained will be accurate, however, the overhead will increase proportionally consuming network resources. Hence, we propose to use surrogate data obtained by means of an autoregressive estimator to reduce the overhead without impacting our deployment algorithm. We show by simulation the efficiency of both algorithms and their performance in terms of deployment time, delay, jitter, and throughput.

Keywords: Routers (computer networks); Ad hoc networks; Mobile robots; Deployment algorithms; Probe packets.

Résumé

En cas de sinistre, les infrastructures de communication peuvent être partiellement ou totalement détruites, ou devenir inefficaces en raison du trafic élevé. Néanmoins, il est nécessaire d'assurer la connexité entre les équipes de secours et le centre de commandement. Par conséquent, des solutions de communication temporaires sont essentielles jusqu'à ce que l'infrastructure soit rétablie. Dans cette thèse, nous nous concentrons sur le déploiement d'une solution de communication appelée réseaux de substitution. Ainsi, nous proposons un algorithme d'auto - déploiement pour permettre aux routeurs mobiles, composant un réseau de substitution, de se répartir pour couvrir la zone cible. Notre algorithme surveille les conditions du réseau pour décider si le routeur doit ou non se déplacer, il règle la position de ce dernier en fonction des informations provenant des nœuds voisins à un saut au moyen de la mesure active, c'est à dire, les paquets sondes. Ces paquets sondes permettent à l'algorithme de surveiller le canal et ses éventuels changements au fil du temps. Nous comparons les différents paramètres pour évaluer la qualité du lien et nous observons le comportement de notre algorithme de déploiement considérant chaque paramètre séparément. Par ailleurs, nous étudions comment la mobilité contrôlée des routeurs affecte la performance du réseau au moyen des simulations.

Afin d'évaluer la qualité de la liaison, la plupart des algorithmes de déploiement utilisent des techniques de la mesure active. Ces techniques nécessitent que les nœuds envoyent des paquets sondes sur le réseau pour obtenir des mesures. La précision de telles mesures dépend de la fréquence des paquets sondes transmis. Cette précision est importante lorsque les caractéristiques du canal changent au fil du temps. Néanmoins, il existe un compromis entre le taux de transmission des paquets sondes et la précision des mesures. Si le taux de transmission des paquets est suffisamment élevé, les connaissances obtenues seront exactes, cependant, le coût augmentera proportionnellement en consommant plus de ressources réseau. D'ailleurs, puisque notre algorithme de déploiement dépend de la mesure de la qualié du lien pour prendre une décision, si on réduit le taux de transmission des paquets sondes, on augmente le temps nécessaire au routeur pour rassembler des informations pour prendre la décision de mouvement et, en conséquence on augmente le temps de déploiement. C'est pourquoi, nous proposons d'utiliser des données de substitution obtenues au moyen d'un estimateur autorégressif pour réduire la surcharge sans impacter notre algorithme de déploiement. Nous montrons par simulation l'efficacité des deux algorithmes et de leurs performances en termes de temps de déploiement, de délai, de gigue et de débit.

Mots clés : Routeurs (réseaux d'ordinateurs) ; Réseaux ad hoc (Informatique) ; Robots mobiles ; Modèles ARCH ; Algorithmes de déploiement ; Paquets sondes.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Events like natural or man made disasters are hard to predict and it is even harder to foresee the damages that such events will provoke. Due to these events, in many occasions the communication infrastructure is destroyed transforming the affected zone into an isolated information island [1]. In addition to this lack of infrastructure caused by physical destruction of the network components, the remaining communication networks, e.g., radio or cellphone networks, may be insufficient due to people calling to emergency services or trying to communicate with their relatives [1, 2].

After a disaster, it is a crucial task to provide connectivity between the rescue team, e.g., firemen, policemen, or paramedic, and their command center so that the first responders report their findings and coordinate the rescue tactics. Therefore, from the networking point of view the goal is to restore and to repair the network in order to provide the communication services. One approach to overcome such a problem is to organize and send out network components, for instance, routers, access points, or relays, in order to replace those that were destroyed or to create a network on demand [3]. This is what we call a rapidly deployable network (RDN).

However, deploying a network under critical conditions presents an important set of challenges. Firstly, the proposed network must be deployed without any *a priori* knowledge about the environment, therefore, its deployment must be done *on-the-fly* as fast as possible, i.e., the network is set up quickly in real time to replace the damaged portion of the infrastructure [1] or to alleviate the traffic congestion. And secondly, the network must be adaptive, flexible, and scalable to deal with unknown dynamic environments, that is, the network is set up on demand accordingly to the necessity of the moment and the place.

Such a problem has motivated a research effort on network components deployment called Rapidly Deployable Networks (RDN). A rapidly deployable network, also known as *impromptu* network or spontaneous network, is an adaptive, mobile and quick distributed communication system [4]. In general, we may enlist the most representative requirements of the RDN design cited in the literature as follows:

Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation. Hence, the network

[1] We call the remaining functional portion of the network as *base network*

must provide and maintain essential services under adverse conditions, as well as, allow a fast full recovery of services [5–7].

Multimedia communications is the transmission of a combination of at least two types of traffic data, video, and audio. It is advisable that the network supports data, voice, and video communications to deliver accurate information transmission from the incident zones.

High bandwidth link is one that may be able to carry enough information to sustain the succession of images in a video presentation. In an emergency, the communications network, e.g., radio or telephone networks, can be insufficient after an incident, caused by people calling to emergency services, so the networks are congested [2, 3]. Thus, RDNs should provide multimedia and data services overcoming saturation.

Linkage of Information the system must interconnect local, regional and international levels.

Quality of Service the network must consider different requirements for QoS depending on the type of traffic.

Self-* self-capabilities, such as, organization, optimization, and healing reducing the human intervention in the network management and increasing the automation of processes [8].

Mobility mobile nodes allow deployment and redeployment of the network. Thereby, the network can be tailored to the incident conditions and improve the network performance [9].

Sánchez et al. (1998) present the pioneering work on rapidly deployable networks for the troops moving constantly in unknown territories [10]. Since then, it is easy to envisage new solutions to rapidly deploy a network thanks to the evolution of the communication devices as well as the evolution of the wireless technologies, particularly wireless networks [11].

Wireless networks (WN) have met a huge success because wireless networks are, in general, flexible, low-cost, robust, can dynamically self-organize, are easy to deploy, and scalable. These characteristics make wireless networks well suited for military, emergency, disaster, and community applications. In particular, an ad hoc network is an interesting solution in disaster relief scenarios because it is made of wireless mobile or static nodes capable of self-organizing and create temporary and arbitrary topologies without any pre-existent infrastructure providing interconnection between nodes [12]. Nowadays, thanks to protocols, such as Bluetooth, ZigBee, Hyperlan, and 802.11x mobile ad hoc networks (MANET) are easy to assemble.

Similarly, wireless mesh networks (WMN) are also a viable solution to rapidly deploy a network. WMNs are packet switched networks with a fixed wireless backbone [13]. WMNs are compatible with already existing technologies since they use the same 802.11 standards. Hence, WMNs have been used as an alternative to the WLAN in order to extend the coverage in public access points and to provide a low-cost connectivity to mobile users [14], a property useful in the context of RDN. The main advantage of using WMNs for spontaneous networking is the inherent resilience achieved through multi-hop communication, the nodes are connected by several links and transmit the data from the source to the destination by relaying the packets through other nodes. Hence, in case of failure, it is possible to chose another path to reach the destination and to avoid disconnections [14–16].

In this thesis, we mainly focus on rapidly deployment networks as the solution to provide communication services in disaster scenarios. We focus on the solutions based on wireless networks, specifically, the substitution networks. A substitution network is a temporary solution to replace a portion of a damaged infrastructure (called hereafter base network) by means of mobile routers (called substitution routers) capable of moving on demand and connected to the base network through bridge routers (Fig. 1.1a) as we describe in detail later in this chapter.

Thus, the context of this thesis is the RESCUE[2] (mobile substitution networks[3]) project. The project's aim is to investigate the underlying mechanisms and the deployment of a network composed of a fleet of dirigible wireless mobile routers. These mobile routers must be capable of using the controlled mobility to alleviate the congestion as well as to create an alternative network and adapting the topology on demand in order to avoid disconnection or degradation of the quality of service (QoS). The project's objectives are three-pronged: 1) definition of a reference architecture for the substitution network as well as the necessary protocols; 2) development of a set for network monitoring and metrology techniques; and 3) implementation of the substitution network following a case study inspired by a real scenario.

System overview

A substitution network mainly consists of two types of nodes that allow the temporary substitution of the damaged components of the base network and the interconnection between the base network and the substitution network:

1. **Bridge routers** that are connected in between the base network and the substitution network, and that are used to forward the traffic from the base network to the substitution network and vice versa (Fig. 1.1).

[2] http://rescue.lille.inria.fr

[3] RÉSeau Coordonné de sUbstitution mobilE

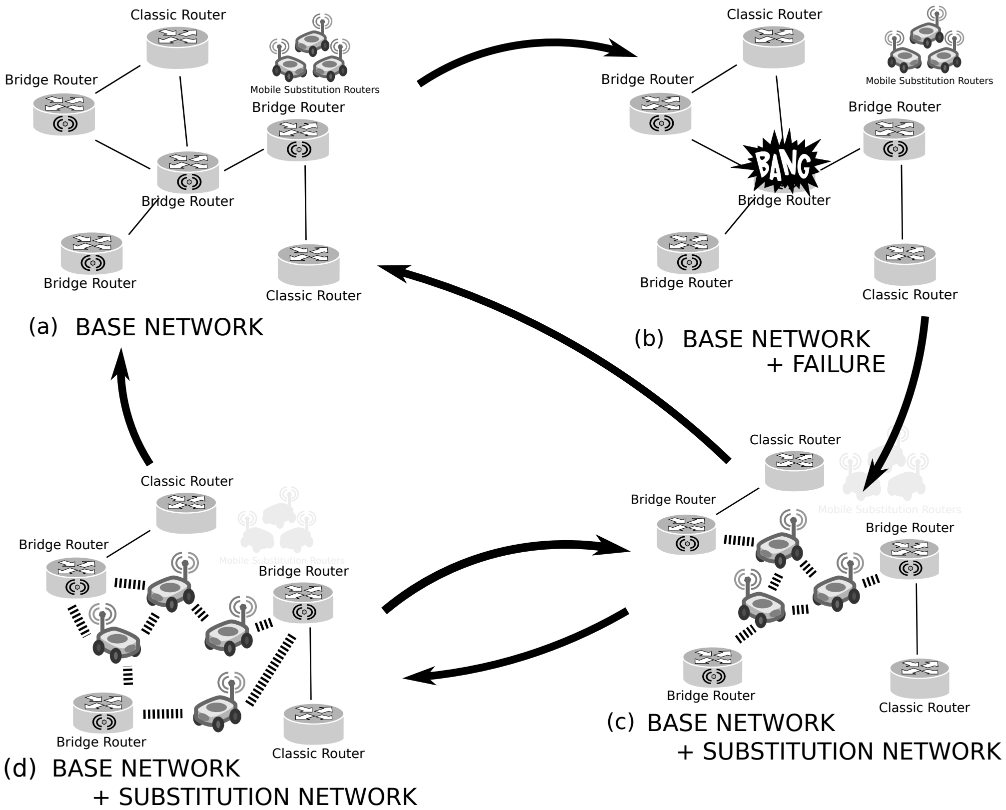


Figure 1.1: Typical use case for a base network and a substitution network. Bridge routers are deployed together with the base network. In case of failure, the mobile routers are deployed or self-deployed to form a substitution network that helps the base network restores basic services such as connectivity.

2. **Mobile substitution routers** that are mobile wireless routers of the substitution network, possibly connected to bridge routers, and whose union provides alternative path(s) to the base network (Fig. 1.1).

The deployment of a substitution network involves the positioning of bridge routers. This placement can be done during the building of the base network or on demand (when extra resources are needed). In order to reduce the human intervention for the deployment of the substitution network, the substitution routers are provided with motion capabilities and a positioning system.

Figure 1.1 depicts the entire perspective of using a substitution network, where the bridge routers are deployed together with the base network (Fig. 1.1a). In this example, the base network operates without the help of the mobile routers. When a failure occurs (Fig. 1.1b), the mobile routers are deployed. In this architecture, the failure detection and the deployment are done autonomously by the base network itself. Mobile routers try to find an optimal position to restore the connectivity service and to ensure QoS to some flows (Fig. 1.1c). In some cases, redeployment may be necessary to improve QoS or to adapt to evolving network conditions (Fig. 1.1d).

Particularly, regarding the routers deployment, our goal is to have an autonomous deployment as well as a possible redeployment of the mobile routing devices. Therefore, it is necessary to design algorithms and protocols to deploy and re-deploy such devices. Since the routing devices are autonomous provided with a limited battery, it is also necessary consider energy constraints during the deployment. Moreover, the deployment computation process does not consider a central entity in the network, hence, this process should be executed in a distributed manner. An efficient router/relay (mobile or static) deployment algorithm must take the link quality into account in order to decide when and where to deploy a relay. For which purpose, the deployment algorithms must be able to measure the wireless link quality.

Contributions of this thesis

A new quality-of-service-based architecture for substitution networks is presented in [17]. This architecture envisions a wireless network composed of mobile routers/relays that provides alternative paths to the base network. Such substitution routers are able to move on demand, so, they can self-deploy and adapt to the network topology accordingly to the environments conditions. Hence, we provide in Chapter 3 a self-deployment algorithm for the mobile routers used in substitution networks. Our algorithm monitors the network conditions to decide whether the router should move or not, adjusting its position based on local information (one-hop information). We compare different metrics to assess the link quality and we observe the behavior of our deployment algorithm considering each metric separately. Moreover, we study how the controlled mobility of routers impacts the network performance through simulations by using the NS-2 network simulator [18] regarding two types of traffic: video traffic and UDP (User Datagram Protocol) traffic.

In order to assess the link quality, most of the deployment algorithms use active measurement techniques. Such techniques require that the nodes send probe packets over the network to obtain the measurements. The precision of such measurements depends on the frequency of the probe packets transmission. This precision is important when the channel characteristics change over time. Nevertheless, there exists a trade-off between the probe packets transmission rate and precision of the measurements. If the transmission rate is too high, important resources like bandwidth and energy will be used in unnecessarily transmission. We may reduce the overhead by reducing the probe packet transmission rate, however, if the rate is too low, the changes in the channel conditions will be not reflected in time. Furthermore, this approach has an undesirable consequence to our deployment algorithm. Since the algorithm depends on the link measurement to make a decision, whether the router must move or not, by reducing the probe transmission rate, we increase the time the router takes to gather enough information to make the movement decision and, consequently, we increase the deployment time. We identify this problem in Chapter 4 and we propose a mechanism based on autoregressive estimation to reduce the overhead without impacting our deployment algorithm.

Finally, we add some remarks about the current work based on the findings presented in this thesis and we conclude in Chapter 5 where we also discuss some perspectives of our future work.

Notations and definitions

We consider a wireless network composed of mobile devices, called hereafter "nodes" that are located and may move on the two-dimensional Euclidean space. For the sake of simplicity, we assume that the transmission range R of a node u is the area in which another node v can receive/send messages from/to u, i.e., $d(u, v) < R(u)$, where $d(u, v)$ represents the Euclidean distance between u and v, and therefore, it exist a link X between u and v.

We assume that two nodes are "neighbors" when they are within the communication range of each other. In the following, we use X_{prev} and X_{next} to refer to the previous and next hop, respectively, of a mobile router. Likewise, we assume that some of the devices are fixed, that traffic needs to be transferred between two fixed devices, and that the wireless routers dynamically move in the scenario and act as relays, regardless of the routing protocol. And, as many link layer protocols, we assume that each node is equipped with a timer and an 802.11 wireless card, as well as with an identifier that is unique in the network (MAC address).

We define the quality of communication link, or just "link quality", as the probability that a message transmitted on the link is successfully received, that is, the reliability of the link [19]. The link quality can be assessed as a function of the received signal strength (RSS) or the signal-to-noise ratio (SNR) [20], for example. In general, higher SNR leads to lower probability of error in the packet. Hence, a link with high SNR is considered a high-quality link [21].

We use the received signal strength indicator (RSSI), signal-to-noise ratio (SNR), round trip time (RTT), and transmission rate (TxRate) as values to measure the link quality because their values retrieve insight of the performance of a wireless network [22]. Therefore, we call RSS, SNR, RTT, and TxRate "link metrics" or "link parameters" in general.

We use the term "broadcast" to refer to the message propagation in a router's neighborhood in order to obtain the link measurements. Also, we refer to the control packets of routing protocols as "hello" messages or beacons and to the packets used in active measurements as probe packets. Finally, we define the term *controlled mobility* as the ability of some nodes to move to specific destination or with a specific goal, i.e., the opposite of randomly [23–25].

Chapter 2
State of the art

Contents

2.1 Introduction

Evans et al. introduced in 1998 the concept of Rapidly Deployable Networks (RDN) [10]. The main idea is to deploy a network infrastructure *in promptu* to provide communication services for military applications. After this work, several deployment schemes have been proposed in the literature focusing primary on emergency communications for connecting first responders, namely, firemen, policemen, or paramedics, with the incident command center. Such a work has motivated a huge research effort on rapidly deployment algorithms, resulting in the following taxonomy: a) femtocell approach [26–28]; b) breadcrumbs approach [29–33]; c) mobile robotic backbone [34–37]. We will discuss each of them in the following.

The femtocell approach consist in setting up one or several home base stations (BSs), i.e., femtocell [38], mainly over a truck, such as ambulances, police cars, or fire trucks. Base stations provides one-hop voice and data communication between the incident command center and the first responders. However, this approach does not offer any opportunity of redeployment or improvement due to the trucks are parked depending on emergency. Figure 2.1 illustrates the femtocell approach, where a single antenna is placed close to the incident command center. The mobile users inside the coverage range are connected to the command center, though, the users that have to go beyond the coverage area are not connected.

We focus in this work on the rapidly deployment networks that consider distributed, inexpensive, adaptive on demand, and on the fly solutions. The deployment algorithms must be capable to deal with environments where the pre-

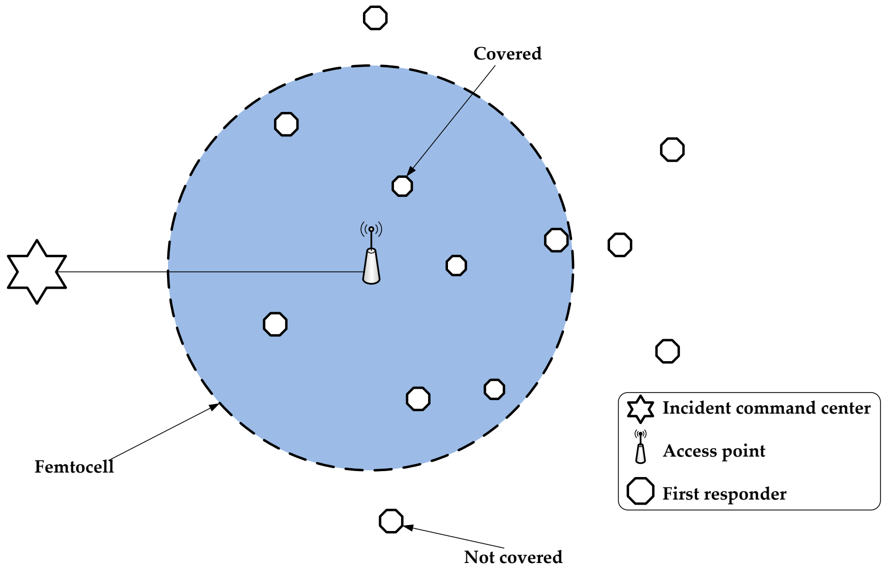


Figure 2.1: A conceptual schema of a femtocell-based network.

deployment mapping is expensive or infeasible without any previous information about the environment. Therefore, in this chapter we focus particularly on two approaches, breadcrumb-based approach and mobile backbone-based approach.

2.2 Breadcrumbs-based approach

The name "breadcrumbs" is a reference to the well-known fairy tale "Hansel and Gretel", where Hansel uses the bread crumbs to trace the way back home. In the context of rapidly deployable networks, breadcrumbs are inexpensive and small devices that act as relays, i.e., their only goal is to forward packets between edge nodes. Thus, following the Hansel example, first responders are provided with several breadcrumbs devices and a mobile radio in a given emergency. The mobile users must drop these devices at regular intervals while exploring the emergency zone to keep the connectivity with the command center. Therefore, the relays are dropped on demand to create a static ad hoc backbone adapting to the environment dynamics. Figure 2.2 depicts an example of a breadcrumb-based network, the command center keeps the communication with the mobile users through relays dropped by the first responders and, thus, enlarging the covered area.

Among the advantages offered by breadcrumb-based approach are that it: 1) creates a multi-hop network on demand; 2) allows communication between the first

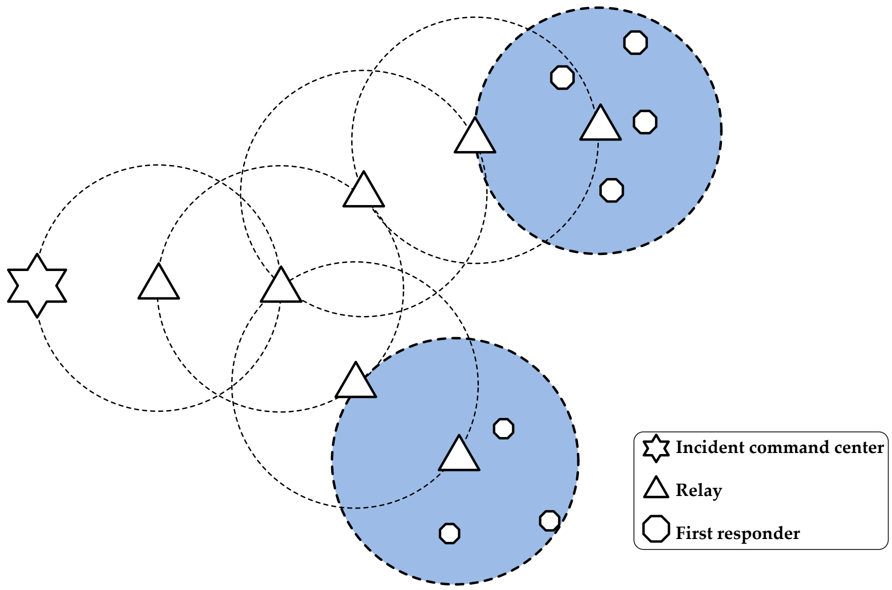


Figure 2.2: A conceptual schema of a breadcrumb-based network.

responders and each other as soon as the relays are deployed; 3) allows communication for isolated relays; 4) guarantees reliable communications; 5) increases the coverage; and 6) reduces the probability of network partitioning. Since the deployment decisions are made in real time without any knowledge about the final topology of the network, the main issue is when and/or where to place the relays in order to optimize the network performance. A naive approach is to establish static and simple rules to drop a relay, for example, any given distance, one relay per floor, or every three doors. However, such a kind of rules only guarantees that the relays are inside the communication range of each other and they do not consider physical phenomena, such as interference, channel fading, background noise, or hidden terminal problem [30, 31]. Thus, the main focus is to propose a deployment decision process that maximizes the network performance.

To address the deployment decision problem, extensive research work has been carried out [29–31, 39–41]. These proposals describe their own deployment algorithms, nevertheless, they share several common points. The algorithms monitor the link quality through received signal strength indicator (RSSI [1]), signal to noise ratio (SNR) or packet loss rate measurements. Such measurements are obtained

[1]There is a significant difference between the received signal strength (RSS) and the received signal strength indicator (RSSI), RSSI is an indicator and RSS is the real value. However, in the papers here cited, the authors use both terms indifferently

by means of probe packets injected artificially in the network or by using directly control messages from the routing protocol. In order to trigger a deployment event, a threshold is set, in other words, once the link quality drops below the threshold, the user must drop a new relay. Recall that the deployment is done manually by the first responder, so, a warning signal to notify the users is implemented as well, for example, a light or a sound. Therefore, an efficient deployment algorithm must be capable of monitoring the network changes to satisfy the relays necessity.

Bao and Lee propose a collaborative algorithm to deploy an ad hoc backbone for spontaneous networks [29]. Such an algorithm takes the link quality into account, for instance, measuring signal to noise ratio or packet loss rate, to decide when to drop a new relay. In order to obtain the measurements, the relays and the mobile users exchange control messages which are added to the control packet header from the routing protocol. The authors assume that each device may differentiate relays from users in its neighborhood. Similarly, they also assume that each user keeps a track of the deployed relays to minimize redundant relays.

The collaborative deployment method introduces two types of control information to request and to announce the deployment of a relay. Thus, a relay must be deployed either because the mobile user detects a link quality degradation or because it receives an explicit deployment request from its neighbors. The latter case occurs when a mobile user runs out of relays, hence, the mobile user sends a request message to its neighbors so that another user drops a relay. Once the deployment decision is made, the candidates to deploy broadcast announcement messages.

In order to decide at which place and at which moment the new relay should be dropped, Souryal et al. propose an algorithm based on a quick evaluation of the physical layer performed by the mobile radio [19,30]. In a nutshell, the mobile radio constantly broadcasts probe packets to the previously dropped relays, when a relay is inside the range, it responds with an acknowledgement packet. Then, the mobile radio measures the RSS through ACK reception, if the maximum RSS value falls below the threshold level, a new relay must be dropped. Additionally, the author propose to use the SNR instead of RSS to measure the link quality, arguing that SNR present lower latency.

Wolff et al. describe in [31] a set of rules to determinate the optimal number of breadcrumbs needed to cover an incident area avoiding undesired effects, such as interference. Their rule is described as follows, the user must drop a relay when the RSS reaches a pre-defined threshold. Such a threshold shall depend on the requirements of each application. They test the deployment rule as well as their own relay prototype by using a realistic scenario composed by network simulation (OMNeT++), a combination of mobility patterns from real mobile users and mobility models, and finally, a 3D mapping from a real scenario.

In the above mentioned proposals, the deployment action depends on the users, i.e., the users have to take a direct action by dropping the devices. However, this is not necessarily the ideal case. When the first responders enter into an emergency zone, their primary activity is not the relay deployment and, therefore, they may omit to drop the relay or just do not realize the deployment signal. To overcome this problem, an automatic breadcrumb dispenser is presented in [41]. Along with the dispenser, an algorithm based on a utility function is proposed. Such an algorithm balances the gain of communication link given by deploying a new relay and the cost of that deployment given the number of remaining relays.

2.3 Mobile robotic backbone approach

Even if the breadcrumb approach has succeeded due to its flexibility and cost-efficiency, its main disadvantage is that once deployed, the relays remain static. That is, if the environment conditions eventually change, the breadcrumb network may not fit to the new conditions. In recent years, an approach akin to the breadcrumb one has been developed. In this approach, the relays are provided with autonomous mobility for self-deployment. Such a capability allows the mobile relays to adjust their own location on demand. This mobile robotic approach arose due to the advances in robotics and automation, however, a review of the robotic technologies is not in this chapter's scope, but rather the deployment algorithms proposed in the literature. Hence, we assume that the deployment algorithms are independent of the robotic technology[2].

As the breadcrumb approach, the aim of a robotic backbone network is to provide communication connectivity to the mobile users but with a minimal human intervention during the deployment. The fleet of mobile relays must be able to organize and to deploy on their own, and, if possible, optimize the network performances. Hereafter, we use the terms robotic backbone network and robot-based wireless relay network indistinctly.

In Figure 2.3, we illustrate the concept of mobile robotic-based network. In a nutshell, the mobile relay devices are placed in an initial position, together or not, and following a deployment algorithm, they will self-spread across the target zone, creating a wireless backbone.

Pezeshkian et al. propose in [34] an initial convoy arrangement, where the relays follow a robot leader one after the other forming a line. Then, the farther relay in the line will stop and convert in a static relay when the degradation in the received

[2]Most of the solutions presented in this section work over the *iRobot Roomba Create* platform, http://www.irobot.com

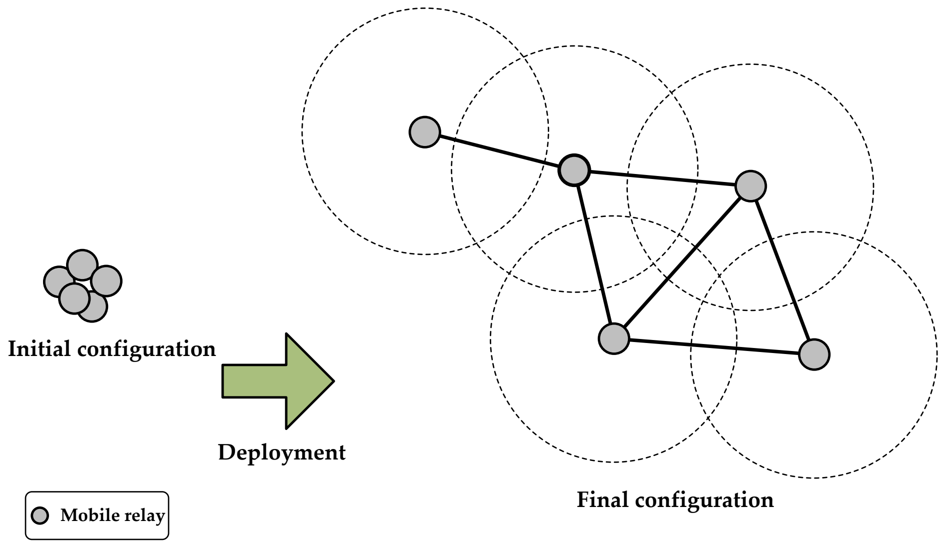


Figure 2.3: A conceptual schema of a mobile robotic backbone-based network.

signal strength crosses a threshold. Such a process is repeated until all the relays become static nodes.

Similarly to the convoy approach, Nguyen et al. investigate in [35] the case of two Access Points (AP) out of transmission range of each other in a wireless mesh network. Thus, a chain of relays is deployed to restore the connectivity between such APs. Their proposed algorithm considers three types of relays: leader, follower, and tail, each relay moving accordingly to its type. All relays are placed close to the first AP. Hence, the relay leader moves forward until it finds the second AP. Each time the value of the received signal strength indicator (RSSI) decays under a given threshold, a follower relay will move to maintain the connectivity between the first AP and the leader, such a follower will follow the node in front creating a chain. Once the leader reaches the second AP, it will stop when it finds the best RSSI value, and the rest of the relays will stop iteratively based on the same rule.

In the aforesaid two algorithms [34, 35], a differentiation between the mobile robots is assumed, i.e., there exist leader robots and follower robots. Such a differentiation may be considered as a weakness because the followers' motion depends on the leader's motion, thus, if the leader fails for any given reason, this would lead to a failure in the rest of the chain.

Timotheou and Loukas report in [36] a robotic-backbone deployment algorithm based on trapped civilians location. Their goal is to maximize the number of connected civilians in order to communicate them with the rescue command center. The robots search for civilians and cooperate to maintain the backbone connectivity. To this end, each robot controls its movement by executing three major steps: exploration, connectivity, and settle. During the first stage, the robot explores the zone searching for civilians; the robot will stop moving if the movement breaks the network connectivity or if the robot provides connectivity directly to the civilians. In the latter case, the robot becomes cluster-head and assigns the exploring task to other robots. One disadvantage of this technique is that the authors assume that the robots have an a priori knowledge of the disaster zone.

In [42], Reich et al. consider a mobile network that automatically maintains its own connectivity by moving constantly its nodes. The authors propose an algorithm to self-spread the mobile nodes over a given area. Each node, moving independently, uses two-hop radius knowledge to determine when to stop its motion accordingly to the decision criterion. The criterion indicates the risk of dividing or disconnecting the network. Thus, each node executes the algorithm, called by the authors SCAN, to maintain the connectivity. The algorithm works as follows: if a given node has a predefined number of link connections with its neighbors, then the node continues moving, i.e., it increases the distances between neighbors; otherwise, the node must freeze because the movement implies a high probability of disconnection.

2.4 Partial conclusion

In order to better understand the context of the potential applications of the substitution networks and their requirements, we have studied the different proposals to rapidly deploy a network, and we have proposed in this chapter a classification for those proposals. We have categorized these solutions in three main branches: femtocell approach, breadcrumb approach, and robot-based wireless relay approach.

Based on the characteristics of each category, we propose to classify the *substitution networking* solution as part of the *robot-based wireless relay approach*. Therefore, hereafter we concentrate our attention on the robot-based wireless relay approach because its attributes match with the requirements of the RDN. Particularly, our goal is to design an efficient self-deployment algorithm. As we have observed throughout this chapter, it is desirable that such a deployment algorithm would adapt dynamically to the environment, for example, different zone sizes, changes in the distribution of mobile users, or changes in the channel conditions.

Chapter 3

An adaptive scheme for the deployment of mobile routers

Contents

3.1 Introduction

It is critical to design efficient algorithms to support pervasive services in networked environments. Some of the main challenges include a fairly complex node placement or to solve a deployment problem without prior knowledge of the optimal network topology or the optimal location of mobile routers. Specifically, we focus on scenarios where end-nodes, such as classical network routers, are disconnected and connectivity with quality of service (QoS) must be restored between them by using wireless routers support controllable mobility, where no backup network exists, and a temporary solution must be quickly deployed [19].

We introduce different adaptive mechanisms for the deployment of wireless mobile routers in pervasive environments. Our solution is:

1. Localized: every decision made by the mobile router is based only on close neighbors (i.e., one hop neighbors) and local link information. The algorithm takes advantage of probe packets to exchange information about their surrounding links status, and thus, it drives the positioning of the wireless mobile router.

2. Scalable: as a consequence of the previous property, our solution is scalable in respect of the network size and the mobility strategy of the surrounding wireless mobile routers.

3. Adaptive: the algorithm ensures that the connectivity quality is permanently monitored based on close neighbors and local link information. As a consequence, the proposed placement scheme is adaptive to topology changes and to the evolution of the network characteristics.

Previous work focuses on the deployment and placement of mobile devices (e.g., robots) for surface coverage [43]. Our goal is to deploy a set of wireless mobile devices placed between classical network routers to restore connectivity and/or to increase the availability of communication service to satisfy application layer requirements, such as delay, throughput, or QoS without prior knowledge of the optimal placement of these devices by using controlled mobility. To that end, we propose an algorithm, which we call *adaptive positioning algorithm*, that dynamically determines the best position, in terms of link quality, of the mobile devices based on local information, and so such mobile devices place themselves at their optimal location.

3.2 Key points

We propose to deploy a network composed of a fleet of dirigible wireless mobile routers for public service. In order to fully adapt to the current conditions, the mobile routers should move or redeploy on demand. This means that, not only the edges of the net may move but also the core or part of the core. One of the deployment issues is in which direction move the router to avoid disconnection or degradation of the QoS.

The deployment of a network composed of a fleet of dirigible wireless mobile routers (named from now on as *substitution network*) can be useful in case of multiple link failures as in natural disasters, weak connectivity, fiber-optic cable cuts, or flash crowds. In this work, we focus on a typical use case of substitution networks as presented in [17]. In this scenario, the substitution network aims at helping a base network to restore and maintain some of the basic services available before the failure. Thus, a fleet of wireless mobile routers is self-deployed to compose a substitution network together with the base network. Thereby, we evaluate an adaptive positioning scheme to deploy a network depending on the driving applications.

Our basic idea is that, during the network lifetime, each wireless mobile device of the substitution network determines a new position by using the feedback on the link quality coming from its neighbors. We use and compare various metrics, e.g., signal-to-noise ratio (SNR), received signal strength (RSS), round-trip time (RTT), and transmission rate (TR) to assess the wireless link quality between a wireless mobile router and its neighbors.

3.3 Proposed algorithm

Our basic idea is that, during the network lifetime, each wireless mobile device of the substitution network determines a new position by using the feedback of the link quality coming from its neighbors. The following are the major steps involved with the algorithm running on each node: (1) measure the chosen link metric, (2) compute the new position, and (3) move to this new position. Each of these steps are described in detail in Algorithm 1, named *adaptive positioning algorithm*, where we use the RTT as link metric.

No prior knowledge of the optimal mobile device locations is assumed to be available at nodes. Our algorithm uses close neighbors and local links information to allow nodes to position themselves. Each wireless mobile router runs the algorithm regularly and measures the link parameters. Next, we briefly describe the steps taken by our algorithm as follows:

1. **Measure link parameters.** In order to measure link parameters, we use an intrusive method. The wireless mobile device regularly broadcasts (every t seconds) *probe request* packets containing a sequence number and the *id* or MAC address of the wireless mobile device. Each node receiving a *probe-request* replies with a *probe-reply message* by using unicast transmission and including information such as its id, its position and any local information regarding the link parameters. We use an intrusive method to get up-to-date information regarding link parameters but also to get a consistent and fair view of each link in the surroundings of a mobile device. An additional advantage of using broadcasting of *probe request* packets is that we can avoid the clock synchronization problem between devices.

2. **Compute new position.** Each node computes its new position based on the surrounding link parameters every $k \times t$ seconds, where k is the number of probe packets to ensure that enough measures are used to get consistent statistics on the link parameter. During the probe period, the wireless mobile device stores the k received values and the measurements obtained through the probe replies. Once the probe period ends, the received values are deleted waiting for new values.

   The wireless mobile device compares the values of the link parameter received from the next and the previous hop, X_{next} and X_{prev}. For example, when the considered parameter is the Round Trip Time ($X = RTT$), if $X_{\text{next}} > X_{\text{prev}}$

then the wireless device will move toward the next node. The degree of the inequality changes according to the link parameter considered. In this case, we assume that RTT is somehow related to the distance between nodes. In case of multiple flows passing through the same device, the wireless mobile device will move towards the node i with the maximum RTT. The link parameter measurements are averaged and used to compute the new position. The mobile device can use different measurements related to the link quality, for example SNR, RSS, or TR. Such values may be obtained by means of any 802.11 wireless card.

3. **Move to new position.** In this step, each wireless device moves forward on the computed direction for a distance d. This stepwise choice is arbitrary and it would be easier to relate the traveled distance d to the link parameter value. However, we choose this stepwise movement to be more realistic since in real environments some geographical positions cannot be considered as a suitable position due to potential obstacles, for example a wireless mobile device cannot cross a vehicles road.

Based on the link parameter measurements, the mobile device tries to equalize the metrics for both the previous node and the next node. It is important to notice here that we assume a correlation between link parameters and position due to wireless channel impairments or fading effects, for example. Figure 3.1 illustrates the major steps of our deployment algorithm. During the probe period, the mobile router sends probe packets to its one-hop neighbors and it waits for the corresponding replies; then, the router, based on the information acquired, assess the quality of each link and decides in which direction will move; finally, the router moves accordingly. This process is repeated throughout the entire time of the simulation.

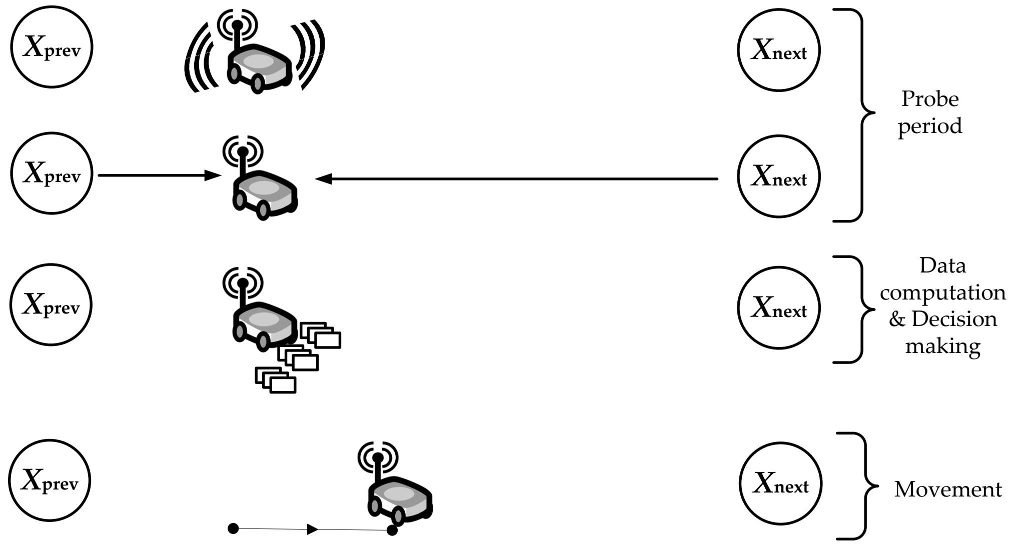


Figure 3.1: General steps of adaptive positioning algorithm.

Algorithm 1 Adaptive positioning algorithm

• **Message formats:**

ProbeRequest messages: Identifier *src*;
ProbeReply messages: Identifier *src*, *dst*;

• **Parameter:**

double ProbePeriod, SendTime, RTT;
int k, Move;

PartI — Link parameters n:

```
set TIMER to expire in time ProbePeriod;
while ( 1 ) do
    if ( TIMER ≤ 0 ) then
        Send ProbeRequest Message;
        SendTime = NOW;
        set TIMER to expire in time ProbePeriod;
while ( 1 ) do
    Upon reception of a ProbeReply
        RTT = NOW - SendTime;
        Store RTT in a table with the ProbeReply sender;
```

PartII — Compute new position and move:

```
set TIMER to expire in time k×ProbePeriod;
while ( 1 ) do
    if ( TIMER ≤ 0 ) then
        Compute link parameter for Next and Prev hops;
        if ( RTT_next > RTT_prev ) then
            Move towards the Next hop ;
        else if ( RTT_next < RTT_prev ) then
            Move towards the Prev hop ;
        else
            Do not move ;
        set TIMER to expire in time k×ProbePeriod;
```


3.4 Performance evaluation

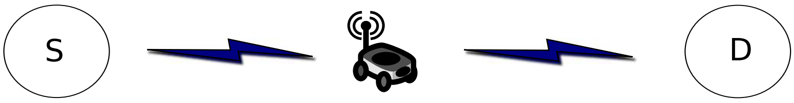
In this section, we present an experimental performance evaluation of our adaptive algorithm. Our goal is to present the effectiveness of the algorithm to deploy wireless mobile routers in a given area. To that end, we evaluate or proposal by using the NS-2 network simulator [18]. Below, we describe the scenarios, simulation parameters as well as the performance metrics we use for the evaluation. We have chosen three scenarios proposed in [44] as result of the study on relay wireless networks.

3.4.1 Simulation scenarios

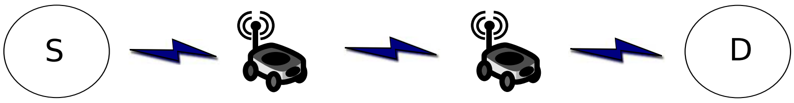
We start by simulating a one-router scenario with a source (`S`) and a destination node (`D`) that communicate through one wireless mobile router as presented in Figure 3.2(a). In this topology, the destination node is placed 250 m far from the source node. At the beginning of the simulation, the router node is placed 10 m far from the source node. Thus, the router starts moving based on our algorithm.

Then, we propose a multiple-router scenario by adding a second router and increasing the distance between the source and the destination to 400 m, the router 1 is placed 100 m far from the source node, and the router 2 is placed 300 m far from the source. This scenario is depicted in Figure 3.2(b). Each router moves independently based on our algorithm with no knowledge about the values obtained by the other router.

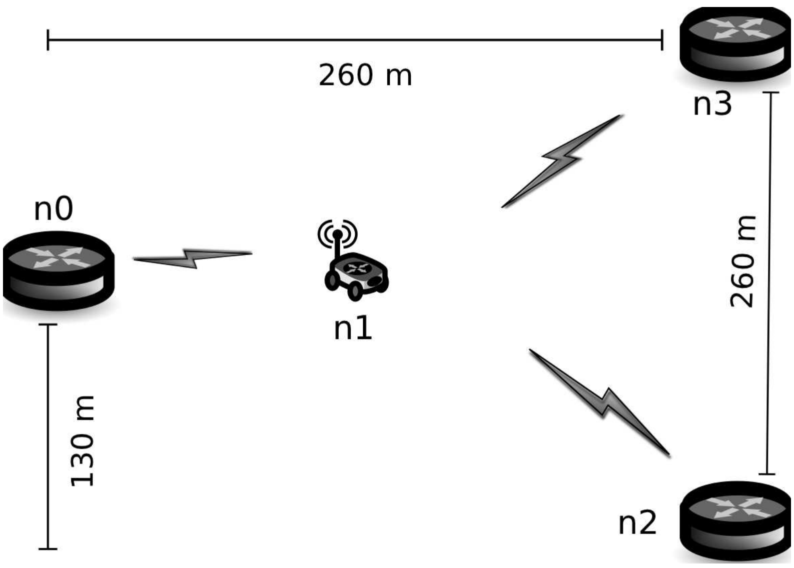
Finally, we evaluate the network performance by varying the number of flows and destinations. We consider a multiple access scenario as depicted in Figure 3.2(c), where there is a source (`n0`) and two destinations (`n2`, `n3`) out of range. So, we use a router (`n1`) to connect the source and the destinations. At the beginning of the simulation, the router (`n1`) is placed 60 m far from the source node (`n0`) on the straight line that connects the source node from the middle position between the receiver nodes (`n2` and `n3`).



(a) Scenario 1: One source, one destination, and one mobile router



(b) Scenario 2: One source, one destination, and two mobile routers



(c) Scenario 3: One source, two destinations, and one mobile router

Figure 3.2: Evaluation scenarios.

We simulate the evaluation scenarios in the well known simulator NS-2.29. We extend the simulator by adding patches that reflect real wireless propagation, physical layer, and the adaptive AutoRate Fallback (AARF) mechanism for 802.11b [45]. AARF adapts the transmission rates depending on the network conditions, in order to increase link reliability. Rather than using a fixed threshold, AARF adapts such threshold following the binary exponential backoff algorithm. We also extend the simulator by adding a realistic channel propagation and error model as proposed in [46], by considering the effect of interference and different thermal noises to compute the Signal to Noise plus Interference Ratio (SINR), and accounting for different Bit Error Rate (BER) to SINR curves for the various codings employed. We use the Dynamic Source Routing (DSR) protocol for our simulations in order to account with an initial routing solution. As we mentioned before, our algorithm is not tied to any routing protocol in particular, so it is designed to work with any routing protocol. Table 3.1 summarizes all the parameters used in our simulations.

3.4.2 Performance metrics

Before we can test our proposed algorithm, we define the set of performance measures we use in our work. In order to assess the performance of our deployment algorithm, we focus on two type of metrics, network metrics and quality metrics that are defined as follows:

- Average throughput ($\mathrm{TH_{avg}}$). The average throughput of a data transfer is: $\frac{F}{\Delta}$bits/sec, where F is the number of bits transferred every second to the final destination during the time interval Δ.
- Instantaneous throughput ($\mathrm{TH_{ins}}$). The number of bits transferred to the final destination in any given instant.
- Average end-to-end delay ($\mathrm{D_{avg}}$). This is the total average time for a packet to travel from source to destination.
- Average jitter ($\mathrm{J_{avg}}$). We compute jitter as the measure of the variability over time of the packet latency across a network; as known, jitter is a function of the delay. The jitter of a packet i is calculated as: $J(i) = J(i-1) + (|D(i-1, i)| - J(i-1))/16$
- Loss percentage ($\mathrm{L_{perc}}$). The loss percentage is equal to $\frac{l-T}{l} \times 100$, where l is the total number of packets sent during the simulation time, and T is the total number of packets received at the destination.
- Peak Signal-to-Noise Ratio (PSNR). PSNR measures the error between any pair of reconstructed and original images. PSNR is usually expressed in terms

Table 3.1: Simulation parameters.

Physical	Propagation	Two Ray Ground
	Error Model	Real [46]
	Antennas Gain	$G_t = G_r = 1$
	Antennas Height	$h_t = h_r = 1$ m
	Min Received Power	$P_{r-thresh}$=6.3 nW
	Mobile Router Energy	50 J
	Communication Range	240 m
MAC	802.11b	Standard Compliant
	Basic Rate	2Mbps
	Auto Rate Fallback	1, 2, 5.5, 11 Mbps
LL	Queue size	50 pkts
	Policy	Drop Tail
Routing	Static	Dijkstra
	Routing Traffic	None
Transport and Application	Flow	CBR / UDP
	Packet Size	1052B
Multimedia	Video	Encoding rate (high)
	Number of Frames	2101
	Duration	70s
Statistics	Number of samples	$k = 10$
	Broadcast period	$t = U(0.1)$
Mobility	Movement step	$d = 2m$

of the logarithmic decibel scale (dB). If the measured PSNR is between 30 and 50 dB, it is considered as a good quality [47].

- Quality of Experience (QoE) using Mean Opinion Score (MOS) [48]. QoE corresponds to the quality perceived by the final user. In order to measure the perceived quality, an original multimedia transmission is recorded under a controlled environment, in other words, without errors or losses; then, this record is sent through out a given network; finally, a group of users compares the received stream with the original one and they give to the stream a note between 1 to 5, see Table 3.2. The Evalvid tool [47] allows us to process the simulation results after the video crosses the substitution network. In [47], the authors propose an approximation to the QoE by converting PSNR to MOS; this is shown in Table 3.2.

Table 3.2: PSNR to MOS conversion.

PSNR[dB]	MOS	Quality
> 37	5	Excellent
31 - 37	4	Good
25 - 31	3	Fair
20 - 25	2	Poor
< 20	1	Bad

3.5 Simulation results

We implement all the variants of the main algorithm described in Section 3.3 and we measure TR, RTT, SNR, and RSS under the parameters presented in Section 3.4.1. We also use the topologies shown in Figure 3.2. In this section we present our findings.

3.5.1 Positioning of one router

We start by implementing a basic scenario composed of one mobile router and two nodes: a source (`S`) and a destination (`D`), which are placed 250 m far from each other as we show above in Figure 3.2(a). At the beginning of the simulation, the router node is placed 10 m far from the source node. Thus, the router starts moving by using our algorithm.

Three UDP flows are sequentially transmitted from the source to the destination. They try to use the total available bandwidth. UDP packets are generated at a rate of 11 Mbps; the size of each UDP packet is 1 MB. Then, we also vary the average transmission rate with steps of 10, 50, 200, 300, 600, and 1000 kbps for each set of simulations. Each simulation runs for a period of 1000 seconds.

Figure 3.3 illustrates the movements of the router between `S` and `D` while trying to reach the best location. The algorithm decides the movement direction according to the link quality (QoS) using TR, RTT, SNR or RSS metrics. The router is initially placed 10 m far from the source. We observe that only with the RSS-based variant, the router reaches exactly the middle point, 125 m from the source.

This is an expected behavior since SNR is computed based only on the power transmission and the propagation model. When the RTT is the chosen link metric, the mobile router reaches the furthest position from the middle point between source and destination. This is due to the fact that even if the probe packets have higher priority in the node's queue than data packets, they cannot preempt an ongoing transmission at the MAC layer.

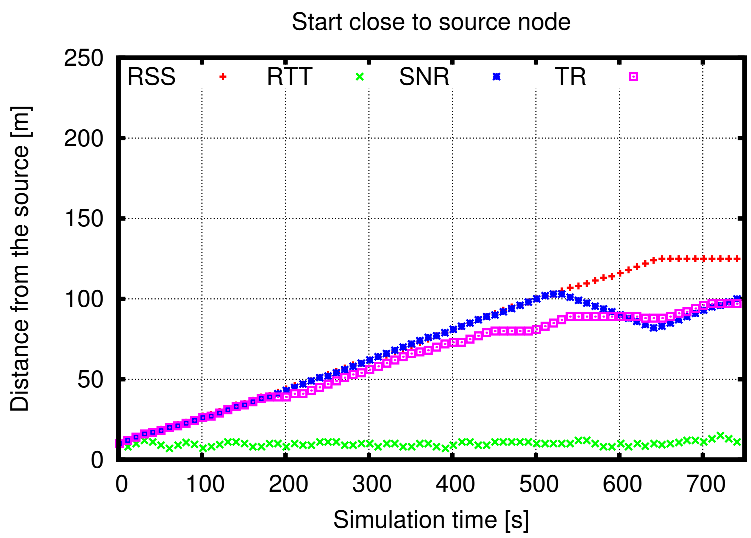
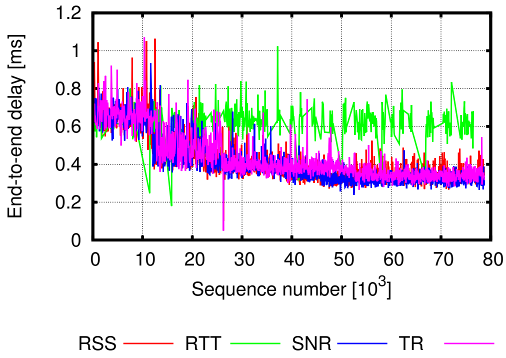


Figure 3.3: Movements over time of the mobile router according to different link metrics.

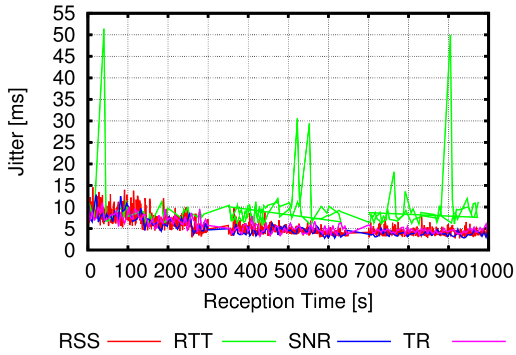
Since there is a high probability that the source node is processing a packet at the MAC layer (being the source of traffic), the RTT between source and router nodes increases for the probe-reply packet. Besides, at the destination node, there is no packet processed at the MAC layer since it does not generate any traffic. Thus, the RTT decreases at the destination node, which explains such behavior for the RTT-based positioning.

It is also worth noting that the TR-based and SNR-based positioning of the router node reaches positions close to the middle point. However, since both variants depend on the network traffic, the movement is not as stable as for the RSS. Indeed, the TR uses auto-rate fallback which is based on the number of retransmissions due to collision or packet errors, and SNR uses the interferences caused by other transmissions.

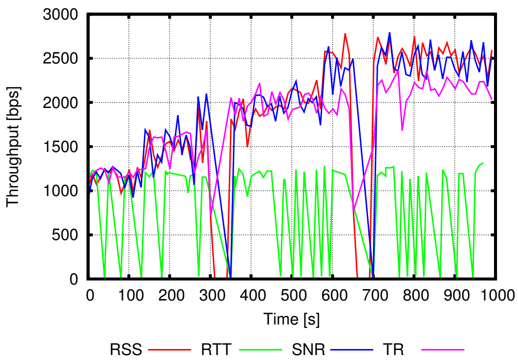
In Figure 3.4, we show the instantaneous throughput, the end-to-end delay for each frame, and the jitter computed as the absolute value of the difference between the delay of two successive packets. We observe that, the worst performance is obtained at the beginning of the simulation, when the router is close to its initial position, except for the RTT variant, which shows stable results around the initial position. In general, once the router starts moving, the performance increases.



(a) End-to-end delay



(b) Jitter



(c) Throughput

Figure 3.4: End-to-end delay, jitter and throughput UDP traffic results for RSS, SNR, TR, and RTT.

Table 3.3: MOS comparison

Video	RTT	SNR	RSS	TR
V_b	3.25	3.56	3.51	3.49
V_e	3.25	3.73	3.71	3.65

Regarding the RTT, we can observe that the algorithm obtains the highest values for delay and jitter. We can also see that the TR, RSS, and the SNR variants exhibit very similar performance regarding delay and jitter. These results show that a good position for the router node is around the point at equal distance between the next hop and the previous hop.

Accordingly, in order to assess the impact of router's mobility on the quality and particularly on the perceived quality we introduce two high quality videos, the first one, V_b, is sent at the beginning of the simulation (∼100 s) and the second one, V_e, is sent at the end of the simulation (∼600 s).

Each video contains 2101 frames and its duration is 70 s. The video packet size is 1052 bytes. The high quality video is encoded at 1024 kbps. We mix the video traffic with a non-saturating UDP flow as a background traffic. The UDP packets are generated at a rate of 400 kbps.

Thus, we plot the metric for the different videos. Figures 3.5(a)(b)(c) depict the results for PSNR, end-to-end delay, and jitter obtained using the RTT, SNR, RSS, and TR variants of our algorithm. We show the PSNR, end-to-end delay, and jitter as functions of video frame number for the two videos. For each metric, the video quality remains higher than the typical values (30 dB), and even in the worst case, the PSNR is higher than the acceptable values (20 dB). The figures show that the first video has lower PSNR, higher end-to-end delay, and higher jitter variation compared to the last video.

Table 3.3 compares the MOS values for both videos. We observe that V_b obtains lower MOS values than V_e for all the variants of the algorithm.

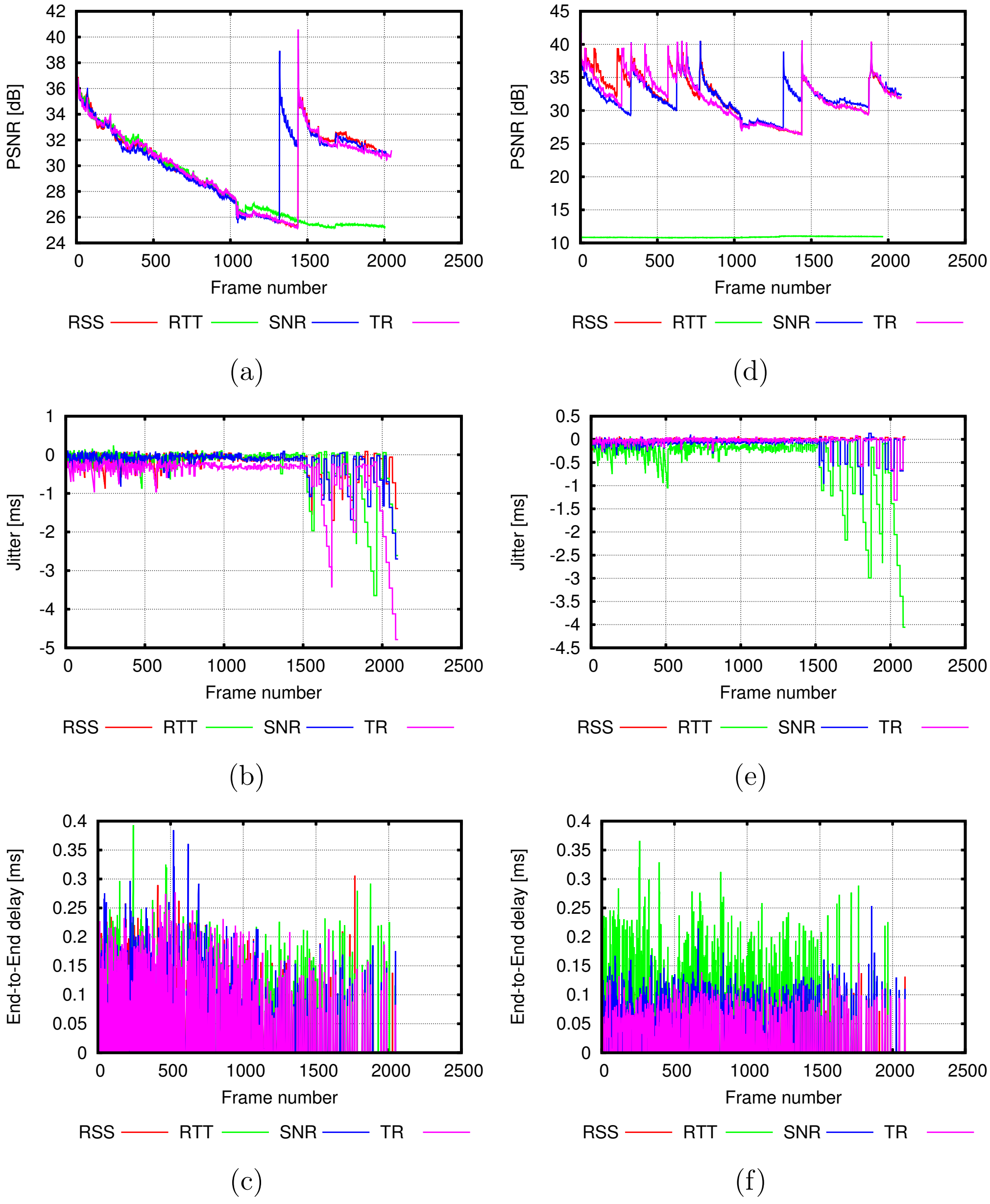
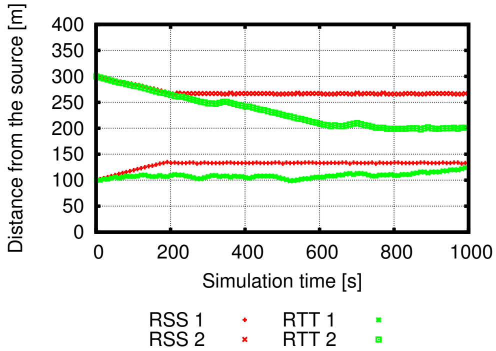


Figure 3.5: High quality video and background traffic results for V_b(a,b,c) and V_e (d,e,f) when RTT, SNR, TR, RSS variants are simulated.

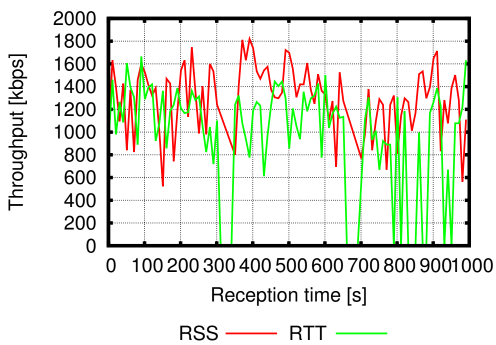
3.5.2 Positioning of two wireless mobile routers

Likewise, we implement the second scenario composed of one source (`S`), one destination (`D`), and two routers (`r1, r2`) as depicted in Figure 3.2(b). We send the same type of traffic as in the previous scenario and we observe the wireless mobile routers' behavior.

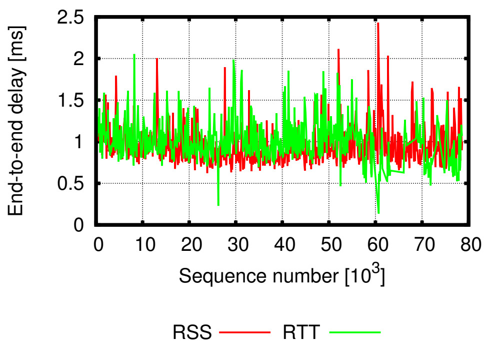
Figure 3.6(a) compares the movement of two routers between source and destination trying to reach the best position based on RTT and RSS metrics. In Figure 3.6(a), router 1 starts at 100 m from the source and router 2 starts at 300 m from the source. We observe that by using the RSS-based variant the mobile router reaches exactly the evenly spaced position between source and destination: router 1 at 125 m from the source and router 2 at 275 m from the source. We can also see that once again the RTT variant of the algorithm shows the worst behavior regarding delay (Fig. 3.6(b)), throughput (Fig. 3.6(c)), and jitter (Fig. 3.6(d)).



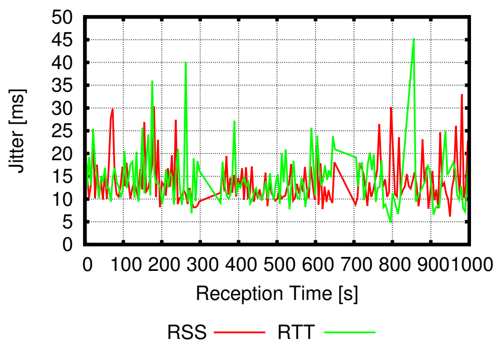
(a) Time plot of two routers



(c) Throughput



(b) End-to-end delay



(d) Jitter

Figure 3.6: UDP traffic (RSS, RTT)

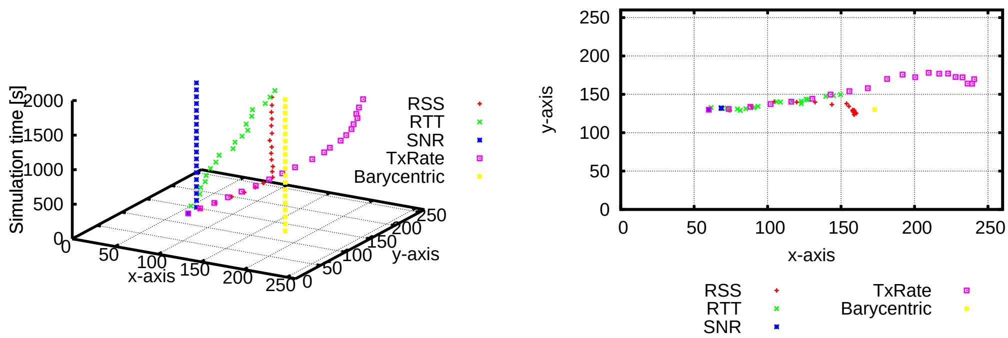
3.5.3 Positioning for two destinations

Finally, we evaluate the network performance changing the number of flows and destinations. We use the topology depicted in Figure 3.2(c), where we present a source (n0) and two destinations (n2, n3) out of range. So, we use a router (n1) to connect the source and the destinations. At the beginning of the simulation, the router (n1) is placed 60 m far from the source node (n0) on the straight line that connects the source node from the middle position between the receiver nodes (n2 and n3). For this scenario, we consider transmitting UDP packets with a size of 1 MB, and we vary the transmission rate as we described in the previous section.

We compare the performance of each link parameter versus the performance of a fixed node. The fixed node is positioned on the barycenter of the given topology, i.e., 173 m far from the source node on the straight line that connects the source node from the middle position between the receiver nodes.

We present in Figure 3.7, the positioning evolution of the router by using our algorithm. This figure presents two views of the evolution, Figure 3.7(a) is a 3D view showing the movement on the Cartesian plane with the time on the z axis. We observe that, when the router uses SNR as the equalizing parameter, it stops moving after 200 seconds, by using RSS, it stops moving after 1500 seconds, whereas by using RTT and TxRate, it continues moving until the end of the simulation. The movement trace is depicted in Figure 3.7(b). Here, we observe that by using TxRate, the router goes close to n3. We also observe that only the RTT parameter reaches the point that is closer to the barycenter, and the RTT based scheme improve its performance in this scenario compared with the simple scenario presented before.

In the following simulation campaigns, we transmit two UDP flows starting at the same time. The source node (n0) transmits Flow 1 to destination node 1 (n2)



(a) Cartesian plane in time 3D. (b) Cartesian plane perspective.

Figure 3.7: Deployment evolution in time (RSS, RTT, SNR, TxRate and Barycentric), contrasting the router placement as function of time.

and Flow 2 to the destination node 2 (n3). In Figure 3.8, we show the average end-to-end delay and the average jitter comparison for each flow. The performance for these two parameters, when the mobile router is positioned on the barycenter, is constant. We see also that for a transmission rate under 600 kbps, the mobile router obtains low values for both delay and jitter.

Figure 3.8 shows the results for throughput and the packet loss. We can see that when the router is positioned on the barycenter, as the transmission rate increases the results are constant and outperform those obtained by using the mobile router. Besides, as the transmission rate increases the packet loss increases. For both metrics, the algorithm performs better when considering the RSS parameter than when considering the other SNR, RTT, or TxRate parameters.

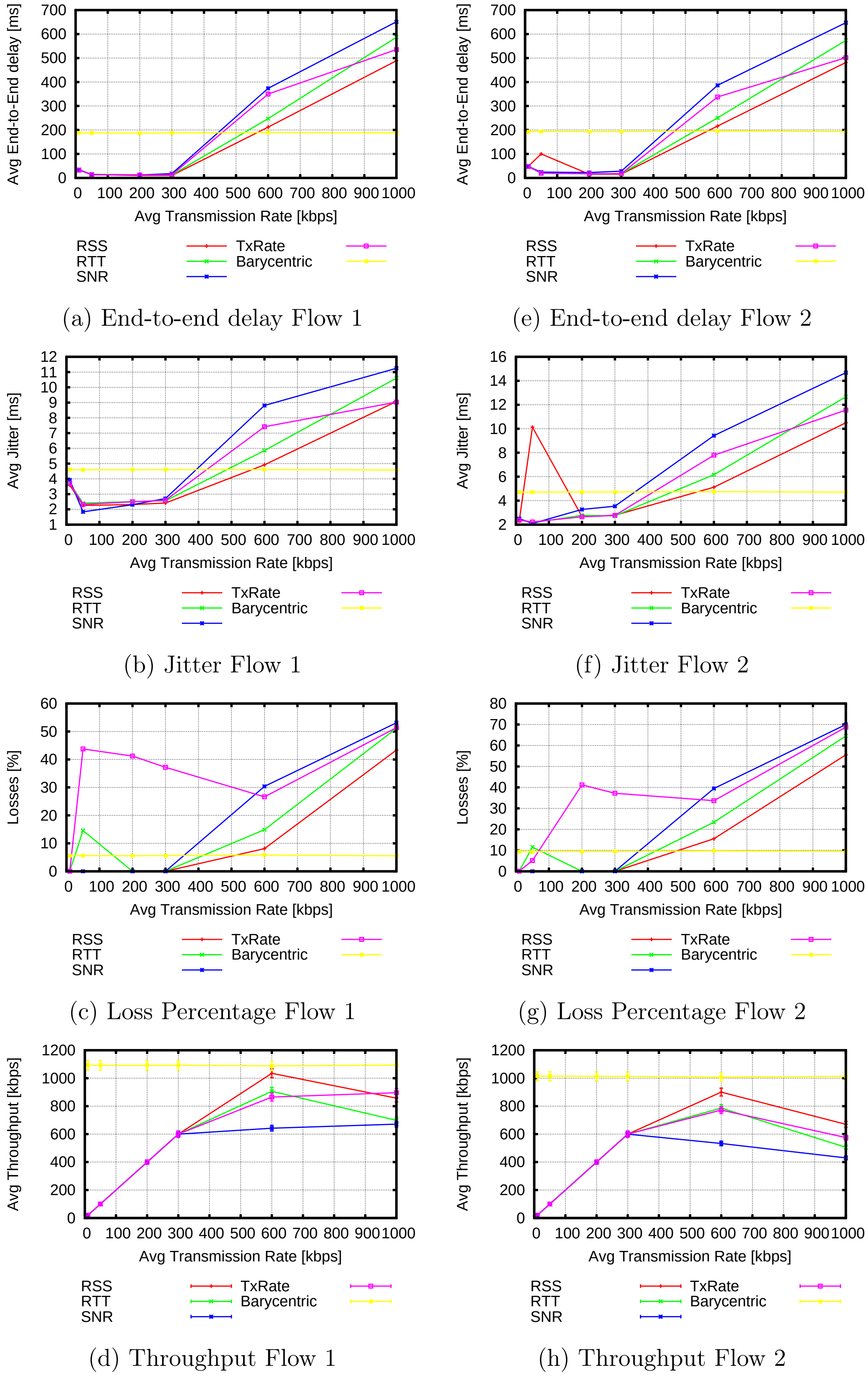
However, these results do not reflect the performance during the simulation time. This is important because, as we can see in Figure 3.9, the mobile router at a given set of times improves the throughput values obtained with the static router. We have to recall that the mobile router starts moving from a position 60 m far from the source node, which is worse than the barycenter in terms of network performance; but, when using our algorithm, the mobile router improves its position and its performance on-the-fly.

3.6 Partial conclusion

In this chapter, we have presented a scheme based on different link parameters for substitution networks that operates in environments where connectivity guarantee is an issue. To that end, we simulated a substitution network with the NS-2 network simulator, testing three different scenarios.

We have introduced a suite of algorithm strategies to control the placement of wireless mobile devices. In particular, we have focused on networks where the source and the destination nodes of UDP traffic are connected through multi-hop communications performed by wireless mobile devices that act as relays. Specifically, our goal has been to *deploy* or *re-deploy* the wireless mobile devices so that application-level requirements, such as data delivery or latency, are met.

The algorithm we have proposed achieves these goals by using a localized and adaptive approach that determines the best positions of wireless mobile routers in terms of delay, jitter, loss percentage, throughput, and PSNR. Our simulation results show the importance of the placement of wireless mobile routers to increase the performance at the application level. Finally, we compared our solution with the optimal theoretical placement, which is the barycenter.



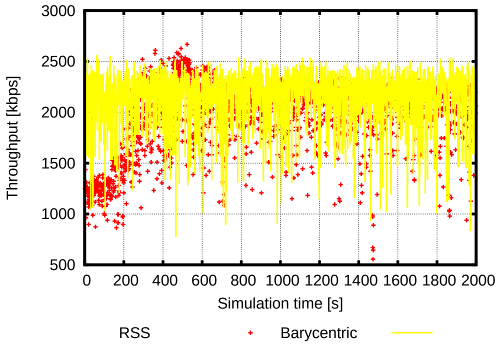
(a) End-to-end delay Flow 1 (e) End-to-end delay Flow 2

(b) Jitter Flow 1 (f) Jitter Flow 2

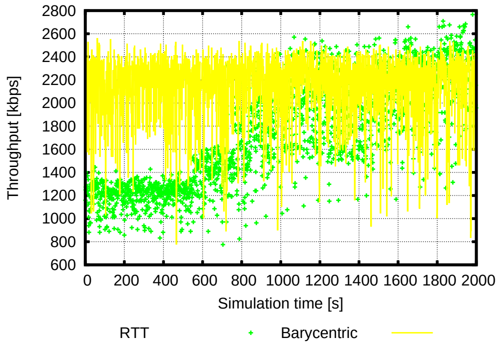
(c) Loss Percentage Flow 1 (g) Loss Percentage Flow 2

(d) Throughput Flow 1 (h) Throughput Flow 2

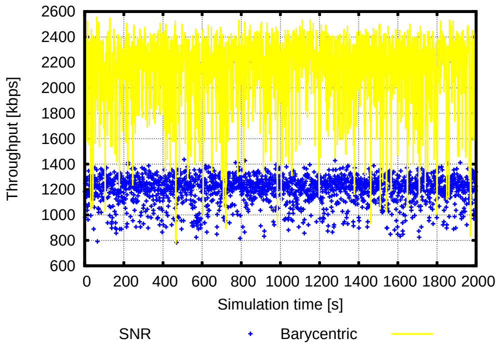
Figure 3.8: End-to-end delay, jitter, throughput and packet loss percentage comparison.



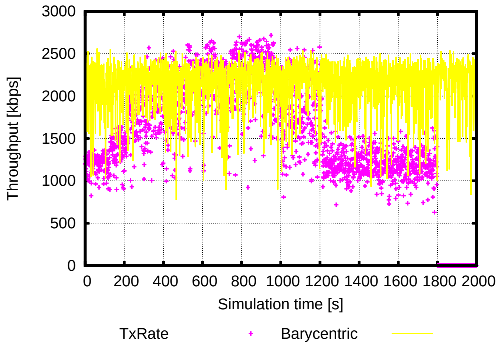
(a) RSS-based adjustment



(c) RTT-based adjustment



(b) SNR-based adjustment



(d) TxRate-based adjustment

Figure 3.9: Instant throughput comparison between each link parameter and barycentric

Chapter 4
Autoregressive estimation for overhead reduction

Contents

4.1 Introduction

We have proposed an *adaptive positioning algorithm* to self-spread mobile routers in substitution networks, presented in Chapter 3. Such an algorithm allows the routers to be deployed by using controlled mobility based on the link quality. The link quality is measured in terms of delay, received signal strength, or signal to noise ratio by means of active measurement. To this end, our proposed algorithm sends probe packets to all the nodes in one-hop communication range; each node that receives the probe packets answers with reply packets; thus, the router computes the next movement based on the replies received.

Active measurement is an effective technique to provide insights about the network performance. Nevertheless, its main disadvantage is that it introduces extra packets, that are *overhead*, to obtain such measurements. Therefore, the accuracy of the measurements depends on the quantity and frequency of exchanged packets. If the probe transmission rate is increased, the insights obtained will be accurate, however, the overhead will increase proportionally consuming network resources. Obviously, there exists a compromise between the information accuracy and the probe packets transmission rate. This compromise is specially important when a decision making process relies on information gathered by using the probe packets such as in the case of the aforementioned *adaptive positioning algorithm*.

In the context of our algorithm, presented in Chapter 3, the compromise between resources consumed and accuracy impacts directly movement decision making process. For example, we may reduce the overhead by reducing the transmission rate, yet, the time the routers take to arrive to the final position is increased too. This is because it takes more time to acquire the necessary data. Thus, the strategy chosen to reduce the overhead must consider this compromise in order to reduce the impact on the router's mobility.

In the literature, we can find some similar problems. For instance, the hello protocol, also known as neighbor discovery protocol, allows neighbor nodes to establish and maintain the neighborhood relationships in wireless ad hoc networks [49]. For that purpose, each node sends hello messages at regular intervals to claim/notify its existence. So, a main issue is to determine the optimal messages transmission rate. Likewise, in wireless sensor networks (WSN) the communication task consumes most of the available energy [50]. Hence, one method to reduce the energy consumption is to reduce the amount of messages exchanged between nodes. The goal of data reduction techniques is, precisely, to reduce the data exchanged between the sink and the sensor nodes. In particular, data prediction techniques reduce the amount of information sent by building a model of the data evolution. In both cases, it is possible to use autoregressive modeling to solve the trade-off issue as presented in [51, 52].

Hence, in this chapter, we propose to use surrogate data obtained by means of an autoregressive model avoiding data starvation periods. Our contribution is two fold; firstly, we compare our proposal with the EEE [52] algorithm and prove its performance with real samples publicly available in order to select the most adequate model. Secondly, we apply the chosen algorithm to the context of substitution networks.

4.2 Overhead and accuracy compromise

The active measurement is a well known approach to characterize a network's conditions and performance. To which end, the active measurement techniques introduce extra packets into the network called *probe packets* [53]. These probe packets provide with insights into network behavior, for example, packet loss, delay, and jitter [54].

Nevertheless, adding packets to the network traffic poses challenges particularly in wireless networks. For example, since the IEEE 802.11b can use several raw data rate namely 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps, the quantity of resources

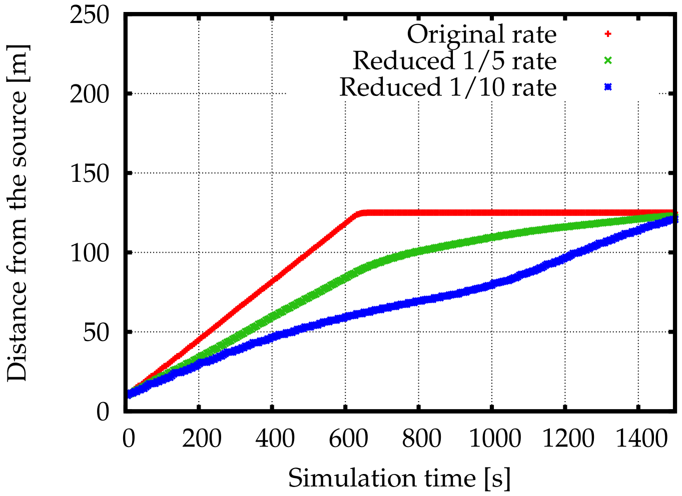


Figure 4.1: Impact of the overhead-accuracy compromise on the adaptive positioning algorithm performance.

consumed by a packet relies on the rate at which the packet is transmitted, i.e., a given packet transmitted at 11 Mbps consumes less resources that an identical packet transmitted at 1 Mbps. Furthermore, it is necessary to consider the additional resources in terms of storage and analysis of the aggregated traffic. Therefore, there exists a trade-off between the amount of probe packets transmitted and accuracy of the obtained values [55]. The above mentioned compromise is specially important when a decision-making process depends on the data collected during the sampling intervals.

Beyond error as an important metric to evaluate the proposals, the ultimate goal is to make the same decision with less information. Particularly, our deployment algorithm needs to gather data about the link quality to decide in which direction the router must move. Such a task depends on the probe packets transmission rate to collect the link quality data. In other words, if the rate is too high regarding the decision making process time, valuable resources, such as energy or bandwidth would be wasted; contrariwise, if the rate is too low, it would cause data starvation periods where the router needs to make a decision but does not have enough information to do it.

To illustrate overhead and accuracy compromise in substitution networks, we consider one source, one destination, one router scenario as presented in Chapter 3.4.1 under the same assumptions. First, we execute the adaptive positioning algorithm without modifications; then, we execute the same algorithm and we reduce the frequency by increasing the time between probe periods, that is, just sending one-fifth of the probe packets total in normal conditions; finally, we increase again the time between probe periods, sending one-tenth of the probe packets. In Figure 4.1, we can observe that, by using the normal rate, the router reaches the middle point after 600 s of simulation. However, when we reduce the probe frequency, the time the router takes to reach the same point increases drastically. Thus, the overhead-accuracy trade-off has an evident impact on our algorithm performance.

4.3 Some background on time series models

Before describing our proposal to reduce overhead, we present a brief introduction to the time series models for a complete understanding of forecasting-based techniques.

Time series forecasting methods are commonly used to predict the output values as a function of previous values of a given series. In particular, the autoregressive (AR) model is widely used due to its simplicity and low complexity. This model predicts the value of X_{i+1}, denoted by $\hat{X}_{i+1}$, and is taken as a weighted sum of the last M values of the process X_i. The AR(M) model is defined as follows:

$$\hat{X}_{i+1} = \sum_{m=1}^{M} \phi_m X_{i-m} + \varepsilon_i, \tag{4.1}$$

where ϕ_m represents the coefficients, M is called the model's order, and ε_i is white noise. There are several approaches to estimate the values of ϕ_m for $m \in [1, \ldots, M]$, such as, Yule-Walker equations, ordinary least squares, maximum entropy estimates, geometric lattice method, and forward-backward method [56]. In practice, the selection of the order's model M for a given data is calculated by balancing the error that the model generates against the number of parameters in the model. It is easy to reduce the error by increasing the order, however, this requires a greater number of unknown parameters, consequently, increasing the resources requirements.

The autoregressive-moving average model (ARMA) and its generalization on the autoregressive integrated moving average model (ARIMA) are a combination of the AR and the moving average (MA) models. In both, the AR branch represents the dependency between the current value and the M previous values, while the MA branch represents the influence of current and past errors due to white noise on the current and future values.

4.4 Autoregressive estimation

In the wireless sensor networks context, data prediction techniques reduce the amount of information sent by building a model of the data evolution [57,58]. Then, the model predicts the future values with a margin of error. The model is built at the sensors as well as at the sink. If the model is accurate enough, the sink will respond to the users queries without the real data; otherwise, the sensor and the sink need to retrieve the actual values to update the model. We believe that the same principle may be used in the substitution networks context as we present later in this chapter.

From now on, we focus only on the algorithm proposed by Ghaddar et al. which we call EEE [52]. Ghaddar et al. in [52] aim to reduce the communication overhead between sensors and their sink by feeding the AR model with its own estimates as samples to generate new estimates. In other words, if $\hat{X}_i$ is close enough to X_i, the model will use $\hat{X}_i$ as sample; otherwise the sensor will send the actual sample X_i and, based on this sample, the sensor and the sink will recalculate the corresponding coefficients. Furthermore, the authors propose a method to dynamically fit the coefficients in Eq. (4.1), rather than using the traditional methods.

The algorithm proposed by Ghaddar et al. is initialized as follows: at time $t = 0$, $\hat{X}_0 = X_0$, $\phi_m = 1/M$ for all $m \in [0, \ldots, M]$, and the estimation error is given by $e_i = X_i - \hat{X}_i$. Each time that e_i exceeds the error tolerance, the ϕ weights must be adjusted sequentially. Generally, the j-th coefficient is adjusted as follows:

$$\phi'_j = \frac{\left[X_i - \frac{j}{j+1}e_i - (\sum_{m=1}^{j-1} \phi_m \hat{X}_{i-m} + \sum_{p=j+1}^{M} \phi'_p \hat{X}_{i-p})\right]}{\hat{X}_{i-j}}, \tag{4.2}$$

where ϕ' denotes the new value of the j-th coefficient $j \in [1, \ldots, M]$. Likewise, the authors use a dynamic error threshold based on previous errors given by

$$\text{thr} = \left(\sum_{i=1}^{M} \left(\frac{|X_i - X_{i-1}|}{M}\right)\right) + \rho, \tag{4.3}$$

where $\rho \in \left[-c\sigma/\sqrt{M}, c\sigma/\sqrt{M}\right]$ is a random number that represents the uncertainty of the estimation due to data dispersion, σ is the standard deviation of the latest errors, and c is the level of uncertainty, in this case $c = 1.96$ for a confidence interval of 95 %.

4.4.1 Tuning the AR estimator

The work shown in [52] proposes an adaptive algorithm to reduce the amount of transmitted data, therefore, saving energy in nodes. This algorithm is based on the AR model adjusting dynamically the model's coefficient with a fixed model's order for all the study cases. The authors use a fixed model's order to simplify their proposal and to avoid taking old meaningless values into account. They claim that the AR(3) model is not efficient since it does not adjust the coefficient values in terms of relative error.

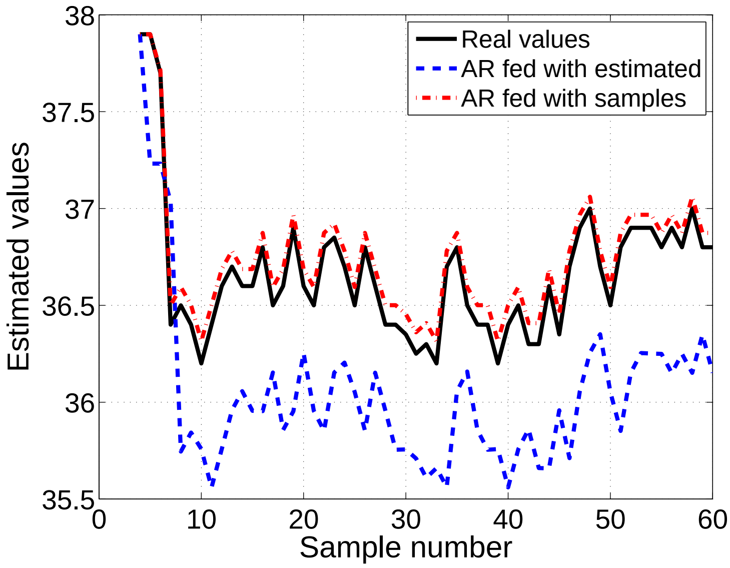


Figure 4.2: The error increases when using estimated values as samples compared with the actual values.

Therefore, we propose to build two AR models based on different methods to fit the coefficients. Considering that we have a set of samples in the past, we calculate the model's order, M as follows. For a given trace, we compute all the values $\hat{X}_i$ for such set of samples, starting with $M = 1$. Then, we increase M by one and repeat the process. The model's order is equal to the lowest value of M preceding an increase in the mean square error. The coefficients ϕ_m in Eq. (4.1) must be fixed in a way that minimizes the mean square error between $\hat{X}_i$ and X_i by using the Yule-Walker and geometric lattice methods [56].

4.4.2 Model feeding

We consider that by feeding the model with its estimates, we reduce the communication overhead, however, this increases the error between estimated and real values. Figure 4.2 depicts this increment, here we compare the actual sampled values with the estimated ones, we use the same model with the same order and the same weights but varying the input to estimate the future values. In order to avoid high inconsistencies between the real and estimated values, it is necessary to add an error threshold. In our approach, each node uses Eq. (4.3) to dynamically calculate a new threshold.

When the threshold limit is exceeded, it is necessary to implement a mechanism to correct the error. Once the threshold is crossed, the estimators use the actual

Table 4.1: Description of the traces.

Trace	Description	Samples	Min-max values
1	Radioactivity in the ground	1441	100–180
2	Daily morning temperature of adult female	60	38–36
3	Carbon dioxide measurements above Mauna Loa, Hawaii	384	315–360
4	Chemical process temperature readings	226	18–28
5	Heart rate measurements	2568	90–190
6	A garden temperature data	1033	14–28

data to correct the discrepancy. In addition, the EEE algorithm re-calculates the coefficients with Eq. (4.2).

4.4.3 Performance comparison

For this first stage, i.e., the validation of our estimators accuracy, we use the value traces obtained from real time series [59]. The trace-based simulation allows the comparison of the algorithms under the very same conditions in a short period of time. A description of the traces is presented in Table 4.1.

4.4.4 Performance measures

In order to evaluate the accuracy of our approach, we focus on the difference between the real data and the estimated data. Particularly, we use the relative error (RE) and the mean square error (MSE). The former reflects the proportional error for each individual value, whereas the latter reflects the overall performance.

Relative error expresses the magnitude of the difference between the real and the estimated values compared to the size of the real values. Recall that X_i is the actual value and $\hat{X}_i$ is its estimation, the relative error is equal to:

$$\mathrm{RE} = \frac{|\hat{X}_i - X_i|}{X_i}. \tag{4.4}$$

MSE is the arithmetic average of the squared errors and is given by:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{X}_i - X_i\right)^2. \tag{4.5}$$

We choose to test the accuracy of the algorithms by means of the numeric simulation based on traces. We assume that each node in the network builds the AR model given by Eq. (4.1). At the beginning of the simulation, the value of `thr` is set to 0.05 and we set the model's order to $M = 3$ for the EEE algorithm as presented originally in [52]. For our proposed algorithm, the sensor collects a set of samples first, and then, it computes the order as we described previously, for example for Trace 1 the algorithm chooses $M = 3$ and for Trace 2, $M = 1$. Each model is fed with its own estimated values as samples.

4.4.5 Results

We compare three different methods to fit the AR model coefficients, first we use the Yule-Walker method, then the geometric lattice method, and finally the EEE method as in [52] by using Eq. (4.2). On one hand, coefficient values calculated by means of Yule-Walker and geometric lattice methods are fixed during the whole session. On the other hand, the coefficients calculated by means of the EEE method change each time that the error exceeds the threshold. The main advantage of the EEE method is that a single simple is needed to initialize the estimator. However, the number of updates required to achieve the coefficients' optimal values is unknown. On the contrary, our AR model takes a finite number of samples, and based on them calculates the coefficients only once.

The results for each trace are presented in Figure 4.3. All the values for relative errors shown in Traces 2, 3, and 4 are smaller than the original threshold value of 0.05. However, we clearly see that Yule-Walker and geometric methods outperform the EEE method, and we confirm this by regarding the MSE values.

To better understand the behavior in the traces, we also present the linear dependence of samples with themselves and two samples in the past. The sample autocorrelation of Traces 1, 2, and 5 is shown in Figure 4.4. Regarding Trace 1, we observe a weak dependence between the samples, therefore, for this case an AR(3) model is more suitable. In contrast, for Trace 5 the correlation is strong over several past samples and a small model's order is advisable. To compare the two previous cases, Trace 2 shows a decay in the correlation, and accordingly the AR(1) works well as seen in Figure 4.3.

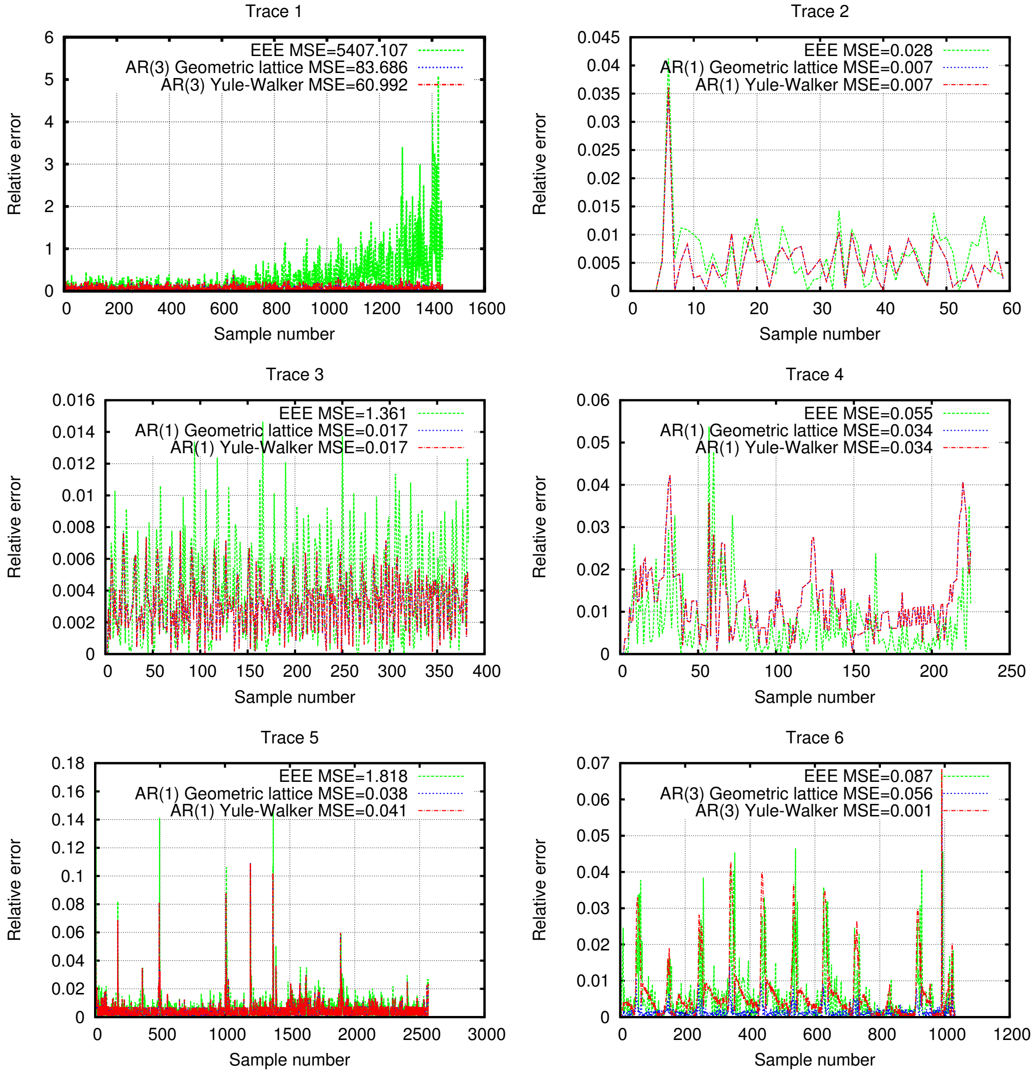


Figure 4.3: The relative error of the estimated values for each trace produced by the Yule-Walker, geometric lattice, and EEE methods. The corresponding MSE is indicated as well.

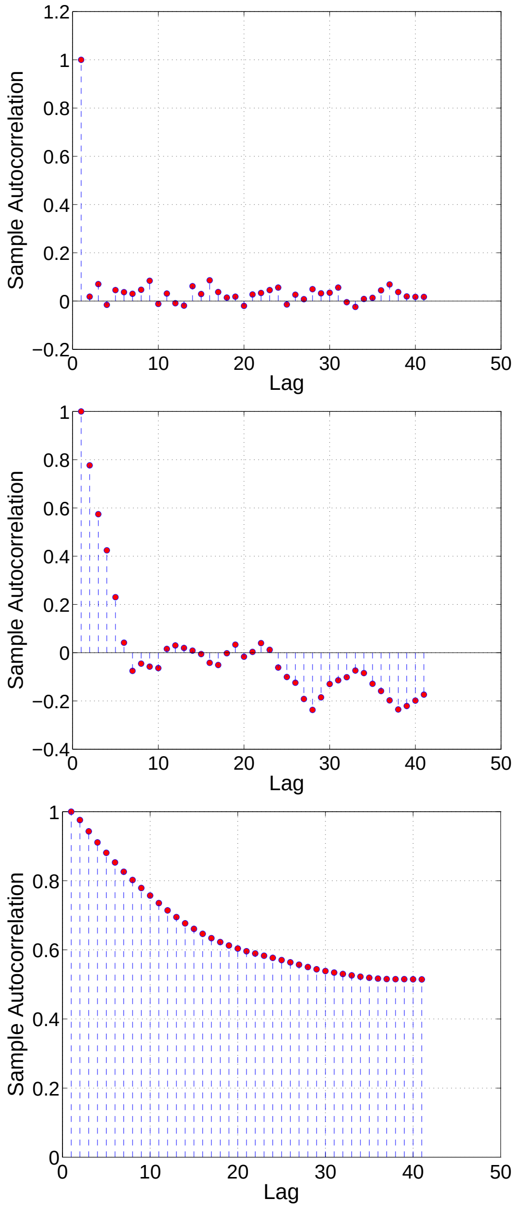


Figure 4.4: Correlation function for Traces 1 (top), 2 (middle), and 5 (bottom), the linear dependence of samples with themselves and two samples in the past.

4.5 Data replacement

In this section, we describe how to apply the autoregressive estimation to reduce the overhead produced by using probe packets. First, we sketch our proposal to reduce the overhead by means of an autoregressive estimation and next, we evaluate the usefulness of our proposal. As we report in Section 4.2, it is important to reduce the number of artificial packets injected into the network preserving at the same time the accuracy of the measurements.

As presented in Chapter 3.3, the adaptive positioning algorithm sends probe packets to all the nodes in one-hop communication range. Then, each node that receives the probe packets answers with reply packets. Figure 4.5 depicts a probe period for two neighbors, the router sends the probe packets in broadcast and receives the replies in unicast. In every probe period, the router sends k probe packets to its neighbors, for example, if the number of neighbors is two and $k = 10$, the router will send 20 probe packets and it will receive 20 reply packets. Of course, this number increases per each additional node in the range and according to the probe periods frequency. It is important to recall that the probe period and decision making period are independent from each other, even though the decision making process relies on data collected by the probe packets (Section 4.2).

4.5.1 Algorithm description

We now describe our autoregressive algorithm for overhead reduction. Since the beginning of the simulation, every router r executes the *adaptive positioning algorithm* and keeps a track of the averages obtained for each link sampled. Those average samples $X_i, i = 0, 1, 2, 3, \ldots$ compose a time series. The algorithm takes as input such time series. Thus, each router constructs one model per each neighbor in one-hop communication range to predict the link quality of its neighbors. We found

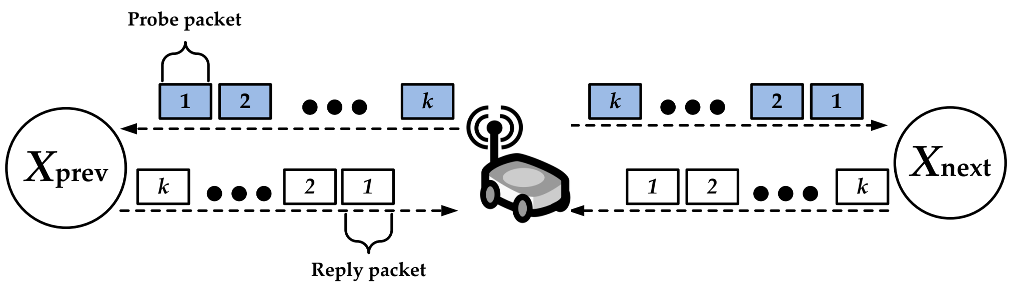


Figure 4.5: Example of a probe period for two neighbors.

that if autoregressive mechanisms are well tuned, important improvements can be achieved on the estimation of real data. Therefore, we choose the autoregressive model as detailed in Section 4.4.1 by using the Yule-Walker method to calculate the model's coefficients. The model's order M corresponds to the value of M that minimizes the mean squared error, MSE, given by Eq. 4.5.

At decision period j, router r estimates the value of the link quality $\hat{X}$ fed with its own estimates until a new probe period brings new data. Then, the algorithm corrects the divergence between real and estimated data by feeding the estimator with real data each probe period. In other words, we use the estimated values to substitute the actual values when the latter is not available due to the overhead reduction. Thereby, the decision making period is not delayed by the reduction of the probe transmission rate. Once the actual data arrives, our estimator uses it to update the model. Figure 4.6 depicts the flow chart of our AR-estimator, router r receives the information gathered throughout the probe period. This information is used for both, to compute the new position and to update the AR-estimator. In case that the information about the link is not available, then, the estimates are used instead.

We describe the major steps of our overhead reduction proposal as follows. At each router r, the average values of the link quality $X_i, i = 0, 1, 2, 3, \ldots$ constitutes a time series. From it, each link in the routers neighborhood is associated to a time series, separately, with individual model's order and individual coefficient values.

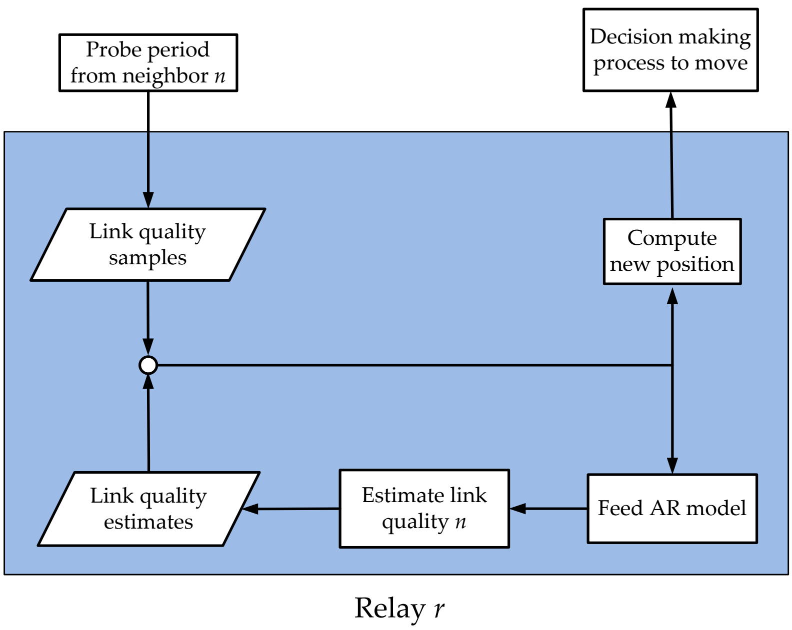


Figure 4.6: Flow chart of the autoregressive estimator at router r.

The estimator collects a set of samples equal to 10, equivalent to 10 probe periods. Thus, at probe period $j > 10$:

- Each mobile router constructs one model for each neighbor in one-hop communication range.
- The estimator is fed with its own samples until the next probe period.
- The estimator retrieves the computed values.
- The router moves accordingly to the estimated values.

4.6 Evaluation

To evaluate our proposed estimator, we use the NS-2 network simulator [18]. We compare our proposal to the original *adaptive positioning algorithm*. First, *adaptive positioning algorithm* is executed without any modifications. Then, we reduce the probe packets transmission rate, and replace the missing data by using estimates generated by means of our autoregressive estimator. We use Equations 4.4 and 4.5 to assess the accuracy of our estimator. For the sake of simplicity, we use the received signal strength (RSS) as an input for both algorithms, nevertheless, it is possible to use other link metric, such as signal-to-noise ratio (SNR) or round-trip time (RTT). We execute 50 simulations of 1500 s for each algorithm in order to obtain average results.

We consider a scenario composed of two nodes, `n1` and `n2`, out of each other range, the distance between them is 250 m. In order to communicate `n1` and `n2`, a router, `r1`, must be deployed. We consider that this router starts at 10 m close from `n1` and it must move to allow the communication between `n1` and `n2` (Figure 4.7). Specifically, our proposal constructs an AR model for each neighbor denoted by X_{prev} (previous hop) and X_{next} (next hop), respectively.

In Figure 4.8, we compare the performance of our proposal, replacing the data with estimates, to the original positioning algorithm that uses the actual values. We show that by using an autoregressive estimation, the router achieves the optimal position after ∼600 s as the original algorithm, i.e., the router make the same movement decision based on the estimated values. Furthermore, we present the relative error for the every sample of the model constructed in Figure 4.9. In both plots,

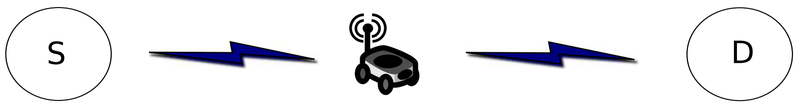


Figure 4.7: Scenario 1: One source, one destination, and one mobile router.

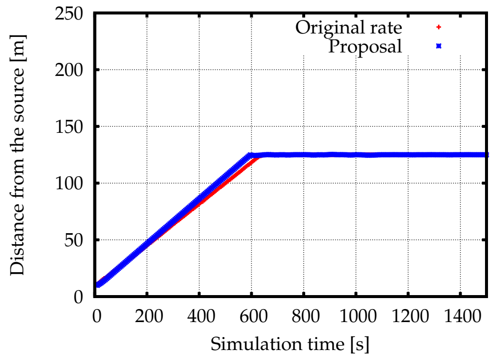


Figure 4.8: Comparing position algorithm performance using actual data and surrogate data as a function of time.

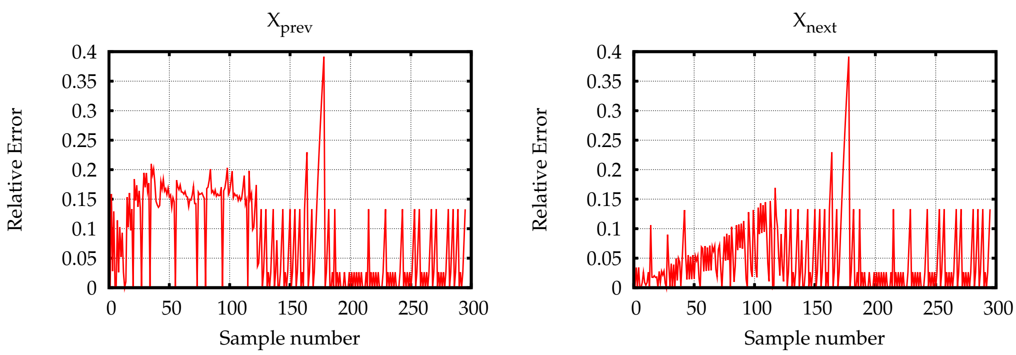


Figure 4.9: The relative error of the estimated values for X_{prev} (previous hop) and X_{next} (next hop) by using the Yule-Walker method as a function of sample number.

the values for relative errors are smaller than 0.4, after the sample number 200, the values of RE achieve the smallest values during the simulation, this corresponds to the moment when the router achieves the optimal position.

Table 4.2 summarizes the gain in terms of number of probe periods and exchanged probe and reply packets. From such a table, we observe a considerable reduction of overhead during the simulation of 80%.

4.7 Partial conclusion

We have shown in this chapter that a well tuned AR estimator may indeed be used to estimate data series in one-hop based information substitution networks. Firstly, we showed how an AR process using the Yule-Walker and the lattice-based approaches both exhibit lower relative errors than the EEE model proposed by Ghaddar et al.

Table 4.2: Packet exchange comparison

	Probe Periods	Probe Pkts.	Reply Pkts.
Without change	428.7	8345.3	8344.14
Reduction to 20%	89.2	1730.86	1729.8
Reduction to 10%	45	863	862.3

The results we obtained for each of the six traces used showed lower relative errors than their EEE counterparts. We also analyzed how autocorrelation functions are of great help when choosing an AR approach. This is clearly an advantage of the algorithms we propose since the autocorrelation function is included when computing the lags for a given estimation.

Later, we focused on this chapter on the use of an AR estimation in wireless substitution networks for overhead reduction. We have reduced the number of probe packets exchanged by 80% meanwhile we maintain the accuracy in the routers movement. In both, the original positioning algorithm and our proposal, the router achieves the same position by expending the same time. Clearly, replacing actual samples with accurate estimates is an effective approach to reduce the overhead caused by the active measurement.

Chapter 5

Conclusion and Perspectives

Contents

5.1 Current work based on this thesis

The findings presented in this thesis has been taken up by other members of our research team. Before concluding, we would like to highlight two extension of our work as a result of the mutual collaboration. Firstly, an improvement to the self-deployment algorithm described in Chapter 3, and secondly, its implementation on a real platform.

5.1.1 Fast-Adaptive Positioning Algorithm

The first contribution is an extension of our deployment algorithm which considers the energy consumption, a fast deployment scheme, and mix of the network metric to (re)deploy the wireless mobile routers [60]. This algorithm, called F-APA (Fast-Adaptive Positioning Algorithm), computes dynamically the distance that the router travels each time that it moves and it combines several link parameters. Recall that in the original algorithm (Chapter 3), the robot travels each time a fixed distance of 1 m. Moreover, the link quality measurement is constrained to one link parameter.

5.1.1.1 Energy consumption

One important contribution is the energy consumption measurement of both deployment algorithms [60]. In order to abide the principles of wireless communications, it is crucial to account with an accurate simulation model. NS-2 applies a linear battery model by default, for this reason we obtain the energy consumed by real mobile devices. Hence, the energy consumption and the speed of the mobile robot

Figure 5.1: Wifibot

is experimentally calculated by using Wifibots [61], and we consider these values for our simulation model.

The energy consumption E_{cons} of the router in Joules is calculated using the following equation:

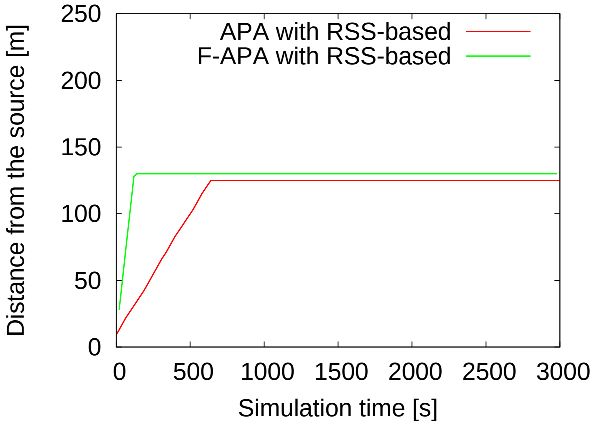
$$E_{\text{cons}} = \begin{cases} 25d + 2 & \text{if the router is moving,} \\ 8 & \text{if the router is not moving,} \end{cases} \tag{5.1}$$

where d is the traveling distance in meters and 2 Joules are added for the robot acceleration. The speed of the robot is 0.9 m/s.

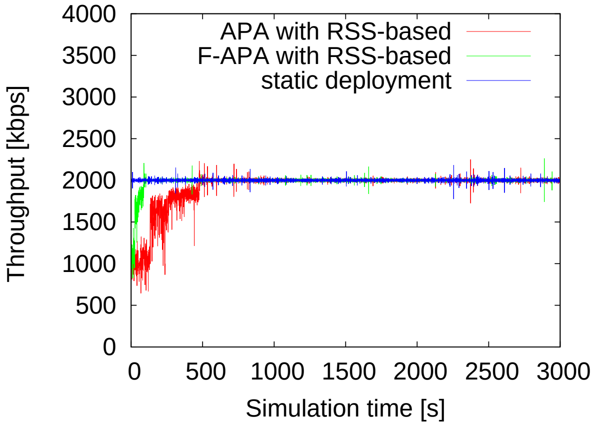
By using the above energy model, we calculate the energy consumed by the robot during the deployment time. The results are presented in Figure 5.2. Figure 5.2(a) illustrates the progress of the mobile relay throughout the simulation using the received signal strength indicator. We can observe that both algorithms reach the middle of the distance, but with F-APA the deployment is faster. The fast deployment also leads to an improved throughput performance (see Figure 5.2(b)) which converges quickly to the rate obtained by the static deployment. F-APA's better performance is achieved by consuming similar energy to APA (see Figure 5.2(c)), the difference is due to robot's acceleration during the first stage of the F-APA algorithm.

5.1.2 Real implementation

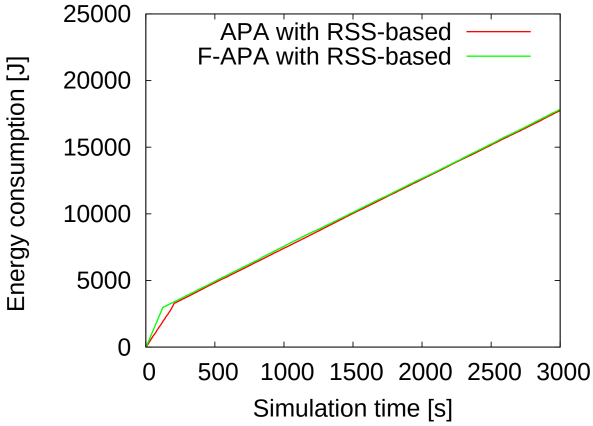
The second contribution is the implementation in hardware of our deployment algorithm by using Wifibots [61] (Fig. 5.1). Such robots are composed of a motion hardware and software and a Atheros Communication wireless card. The scenario at hand consists of one robot (mobile node) and two laptops, source (`S`) and destination (`D`), respectively. `S` and `D` are placed 18 meters apart in a corridor and the initial position of the robot is 1 meter away from `S`. Then, the robot executes the deployment algorithm using the RSSI as link parameter. Due to the constant variation of the RSSI, a movement threshold is also considered otherwise the robot



(a) Distance



(b) Throughput



(c) Energy consumption

Figure 5.2: Received signal strength indicator

never stops. We vary the threshold to 3 dBm, 5 dBm, and 10 dBm to observe the effect of the threshold in the robot's movement.

The preliminary results are shown in Figure 5.3. On the left column we observe the distance that the robot travels each time it executes the algorithm. On the right column, we observe the RSSI mapping according to the position of the robot from both links. Finally, each row corresponds to the threshold values 3 dBm, 5 dBm, and 10 dBm, respectively. For the three thresholds, we can observe the evolution

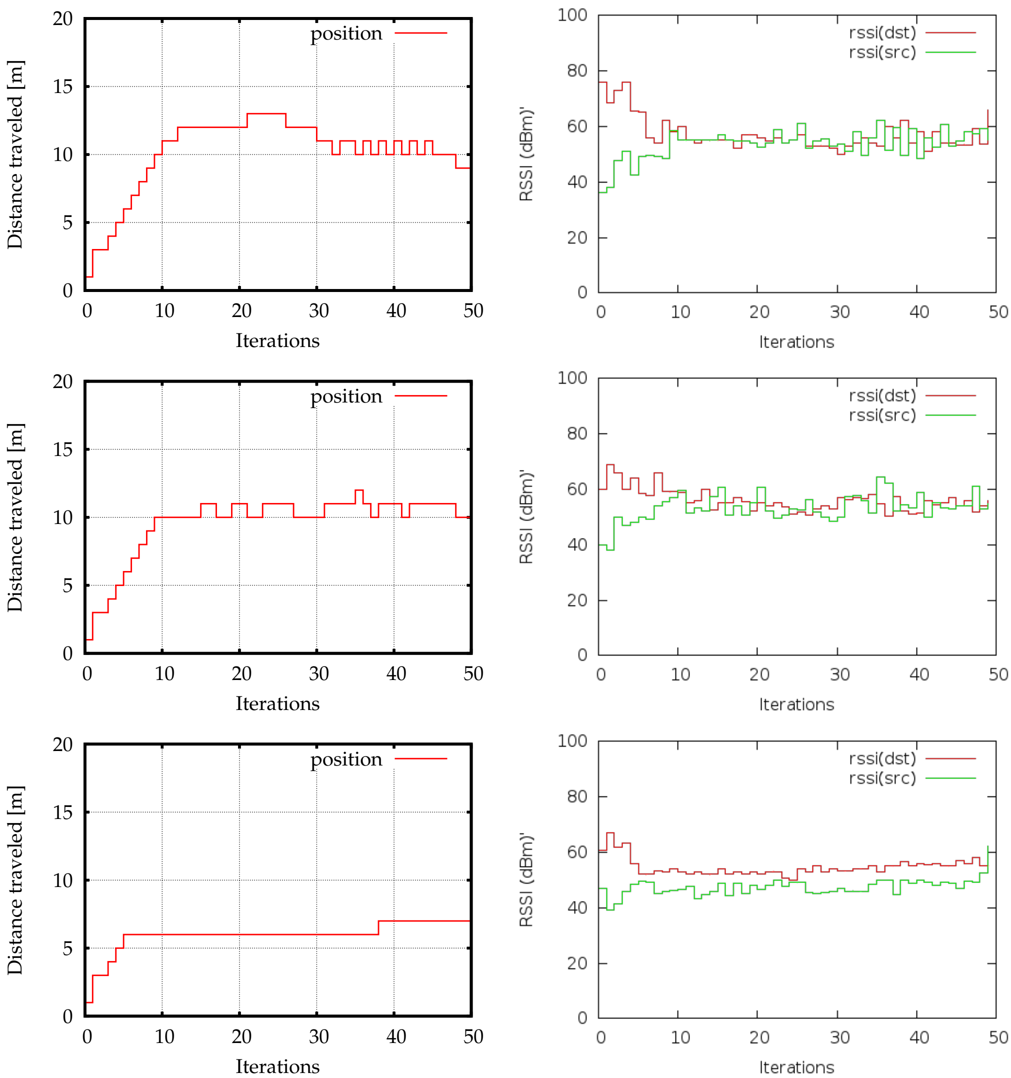


Figure 5.3: The movement of the robot and the RSSI measurements. The robot moves according to the measurement of the received signal, hence we can also map the robot movement to the received signal strength from the two stations.

of the received values tending towards the equalization between link values. These results seem promissory, nevertheless, there still exists a lot of work remaining before having a completely functional platform.

5.2 Conclusion

In this thesis, our goal has been to design and to propose an algorithm to self-deploy mobile wireless routers for substitution networks considering a disaster scenario where part of the infrastructure is damaged. We have been looking for an adaptive, localized, and distributed solution easy to implement and able to deal with environment's evolution.

First, to better understand the context of the potential applications of the substitution networks and their requirements, we have proposed in Chapter 2 a classification for the rapidly deployable networks solutions. This classification is based on the type of wireless technology used to create on demand a network backbone. We have identified three major categories: the femtocell approach, the breadcrumbs approach, and the mobile robotic approach. And finally, we have classified the substitution network solution as part of the robotic network approach.

Second, we have presented a self-deployment algorithm for mobile routers in substitution networks in Chapter 3. Our algorithm controls the routers placement based on the link metrics, such as received signal strength indicator, signal to noise ratio, or round trip time. Our algorithm allows the deployment or redeployment of the routers considering only local information (one-hop) and it is executed in every router independently, i.e., adaptive, localized, and distributed. Moreover, we have evaluated our algorithm by means of the NS-2 simulator, testing three basic topologies. We have assessed the performance of the proposed algorithm in terms of delay, jitter, loss percentage, throughput, and peak signal to noise ratio by using two different types of traffic User Datagram Protocol traffic and video traffic.

Finally, we have proposed an improvement to our deployment algorithm by reducing the overhead caused by using probe packets as technique to measure the link quality. We have used surrogate data obtained by means of an autoregressive estimator in order to avoid data starvation periods due to the reduction of probe packets. To that end, we have compared two autoregressive estimators, the EEE algorithm proposed by Ghaddar et al. in [52] with a fixed model's order and an autoregressive algorithm with a tunable model's order. Then, we have applied the latter algorithm to reduce the overhead by using accurate estimates. Thus, we have reduced the number of probe packets exchanged meanwhile we maintain the accuracy in the router's movement.

Regarding the proposals for self-deployment nodes in rapidly deployable networks, it is clear that it remains a lot of work to do. In particular, an interesting direction is to compare different deployment algorithms for mobile robotic backbone under the same assumptions. This is not a trivial task since, as we see in Chapter 2, each proposal considers different metrics, such as coverage, number of nodes, devices connected, delay, or throughput. And they consider different assumptions about the previous knowledge acquired for the deployment. We think that the careful election of the performance metrics is the next step to continue in this direction. Once the metrics will be defined, we will compare several deployment algorithms initially by means of simulations, and if it possible by means of real implementation.

Regarding our autoregressive mechanism for data reduction in Chapter 4, we have chosen an arbitrary percentage, reducing in 80% the number of probe packets exchanged, in order to evaluate the accuracy of our proposal. Moreover, we also have chosen a gathering information period of 10 probe periods to calculate the estimator coefficients. Nevertheless, these values may be not optimal, so, our future work in this direction will focus on finding the optimal values for the percentage reduction and gathering period. We will evaluate different values and will observe the impact on our mechanism.

5.3 Perspectives

Throughout this thesis, we have focused on the rapidly deployable networks and how to deploy a wireless network, specifically. Nonetheless, a broader view is to consider a *rapidly deployable system* for emergency situations. In such a context, the rapidly deployable networks are only a part of the whole. A rapidly deployable system includes: a monitoring tool before, during, and after a disaster; a public alarm system to prevent the population of possible natural disasters, for example, hurricanes or earthquakes; communication infrastructure in disaster recovery, like the rapidly deployable networks; communication infrastructure for information dissemination and exchange between the disaster zone and elsewhere; a system tool to process and analyze the information about the disaster; and finally it may includes governmental regulations [1, 2, 9]. Due to its characteristics, we think that our contribution may be employed under this system overview.

The deployment of wireless mobile routers is not restricted to disaster scenarios. We think that it may be useful to contrast a surge in the traffic that causes the network to be virtually unreachable, for example, due to a failure of an equipment or a power outage. In these cases, the deployment of a set of wireless mobile routers may help to restore some services to the subscribers. A specific example of an application of a substitution network is the contractor's mistake in the Sydney's Business

District [62]. In 2009, some contractor cut through 10,000 of Telstra company copper wires and 8 fiber-optic cables by mistake. This caused over 12,000 business and residential costumers without phone, mobile, or Internet services for several days. The cost to Telstra of this mistake was AU$1 million just to repair the wires, plus the compensation cost for the affected costumers and a demand by the Australian government. Finally, it took Telstra about a week to replace the cables and restore the service.

We can go even further and consider other scenarios where a substitution network could be useful. We may think about rural or remote communities which suffer from a lack of infrastructure for low-cost access to the Internet [63]. In such scenarios it would be feasible to provide a few high-speed access point and extend on demand the coverage through substitution routers across the communities. This is not a trivial application if we consider that two-thirds of the population does not yet have Internet access. In this direction, we can cite as an example the Project Loon from Google [64], which aims to create a network of balloons traveling on the air to provide Internet access in rural areas.

Lastly, thanks to the ubiquity of the mobile devices, such as cellphones, laptops, or tablets, new perspectives about this devices usage in disaster scenarios have arisen. In a recent paper Iera et al. study the feasibility of using smartphones as gateways/routers in wireless mesh networks [65]. Hence, such smartphones may be used in disaster scenarios to create a wireless ad hoc network not only for the first responders but also for the people trapped in the disaster zone. Then, we may focus on a post-disaster recovery scenario and we evaluate the effectiveness of using commonly available devices to support device-to-device communication.

Publications

Journal

1. Karen Miranda, Enrico Natalizio, and Tahiry Razafindralambo. **Adaptive deployment scheme for mobile relays in substitution networks**. *International Journal of Distributed Sensor Networks (IJDSN)*, 2012

Conferences

1. Karen Miranda, Enrico Natalizio, and Tahiry Razafindralambo. **On the impact of router's mobility on substitution networks**. In Proceedings of the 10th ACM International Symposium on Mobile ad hoc Networking and Computing - MobiHoc (Poster session), pages 3-4, Paris, France, May 16–19 2011

2. Karen Miranda, Enrico Natalizio, Tahiry Razafindralambo, and Antonella Molinaro. **Adaptive Router Deployment for Multimedia Services in Mobile Pervasive Environments**. In Proceedings of the Work in Progress session at IEEE Pervasive Computing and Communication Conference (WIP of PerCom), pages 471–474, Lugano, Switzerland, March 19-23 2012.

3. Karen Miranda, Victor M. Ramos R., and Tahiry Razafindralambo. **Using efficiently autoregressive estimation in Wireless Sensor Networks**. In Proceedings of the 2nd International Conference on Computer, Information, and Telecommunication Systems (CITS), Piraeus-Athens, Greece, May 7-8 2013.

References

[1] Y. Ran, "Considerations and suggestions on improvement of communication network disaster countermeasures after the Wenchuan earthquake," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 44–47, Jan. 2011. 1, 58

[2] A. M. Townsend and M. L. Moss, "Telecommunications Infrastructure in Disasters: Preparing Cities for Crisis Communications," Center for Catastrophe Preparedness and Response, Tech. Rep., Apr. 2005. 1, 2, 58

[3] K. Mase, "How to deliver your message from/to a disaster area," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 52–57, Jan. 2011. 1, 2

[4] J. Evans, G. Minden, K. S. Shanmugan *et al.*, "The Rapidly Deployable Radio Network," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 689–703, Apr. 1999. 1

[5] J. P. G. Sterbenz, R. Krishnan, R. Rosales Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao, "Survivable mobile wireless networks: issues, challenges, and research directions," in *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, USA, Sep. 2002, pp. 31–40. 2

[6] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, and N. R. Mead, "Survivable network systems: an emerging discipline," Carnegie Mellon University, Tech. Rep., 1997. 2

[7] L. Xie, P. Smith, M. Banfield, H. Leopold, J. P. Sterbenz, and D. Hutchison, *Towards Resilient Networks Using Programmable Networking Technologies*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 83–95. 2

[8] S. S. Pundamale, "Survivable networks," Department of Computer Science, University of Helsinki, Tech. Rep., 2007. 2

[9] J. Pancharda and J.-P. Hubau, "Mobile Communications for Emergencies and Disaster Recovery in Developing Countries," National Center of Competence in Research, Mobile Information & Communication Systems, Tech. Rep., Mar. 2003. 2, 58

[10] R. J. Sánchez, J. B. Evans, G. J. Minden, V. S. Frost, and K. S. Shanmugan, "RDRN: A Prototype for a Rapidly Deployable Radio Network," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, pp. 15–22, Apr. 1998. 2, 9

[11] F. Legendre, T. Hossmann, F. Sutton, and B. Plattner, "30 Years of Wireless Ad Hoc Networking Research: What about Humanitarian and Disaster Relief Solutions? What are we still missing?" in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief (ACWR)*, Amritapuri, Kollam, Kerala, India, Dec. 2011, pp. 217–217. 2

[12] C. E. Perkins, *Ad-hoc Networking.* Addison-Wesley, 2001. 2

[13] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations," *Multimedia Tools and Applications*, vol. 29, pp. 285–303, Jun. 2006. 3

[14] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, Mar. 2005. 3

[15] R. Bruno, M. Conti, and E. Gregori, "Mesh Networks: Commodity multihop Ad hoc Networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, Mar. 2005. 3

[16] A. Yarali, B. Ahsant, and S. Rahman, "Wireless mesh networking: A key solution for emergency & rural applications," in *Proceedings of the 2nd International Conference on Advances in Mesh Networks (MESH)*, Athens/Glyfada, Greece, Jun. 2009, pp. 143–149. 3

[17] T. Razafindralambo, T. Begin, M. Dias de Amorim, I. Guérin Lassous, N. Mitton, and D. Simplot-Ryl, "Promoting quality of service in substitution networks with controlled mobility," in *Proceedings of 10th International Conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW)*, Paderborn, Germany, Jul. 2011, pp. 248–261. 5, 18

[18] NS, "Network Simulator v.2 (NS-2)," accessed on January 2013. [Online]. Available: http://isi.edu/nsnam/ns/ 5, 23, 49

[19] M. R. Souryal, A. Wapf, and N. Moayeri, "Rapidly-Deployable Mesh Network Testbed," in *Proceedings of the Global Communications Conference (Globecom)*, Honolulu, Hawaii, USA, Nov. 2009, pp. 1–6. 6, 12, 17

[20] N. Baccour, A. Koubaa, H. Youssef, M. Ben Jamâa, D. do Rosário, M. Alves, and L. Buss Becker, "F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks," in *7th European Conference on Wireless Sensor Networks (EWSN)*, Coimbra, Portugal, Feb. 2010, pp. 240–255. 6

[21] S. Farahani, *ZigBee Wireless Networks and Transceivers.* Newton, MA, USA: Newnes, 2008. 6

[22] E. Feo Flushing, J. Nagi, and G. A. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *International Conference on Computing, Networking and Communications (ICNC)*, Maui, Hawaii, USA, Jan. 2012, pp. 137–143. 6

[23] A. Kansal, M. Rahimi, D. Estrin, W. J. Kaiser, G. J. Pottie, and M. B. Srivastava, "Controlled mobility for sustainable wireless sensor networks," in *Proceedings of the First IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, CA, USA, Oct. 2004, pp. 4–7. 7

[24] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards mobility as a network control primitive," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, Roppongi Hills, Tokyo, Japan, 2004, pp. 163–174. 7

[25] E. Natalizio and V. Loscrì, "Controlled mobility in mobile sensor networks: advantages, issues and challenges," *Telecommunication Systems*, vol. 52, no. 4, pp. 2411–2418, Apr. 2013. 7

[26] D. Câmara, C. Bonnet, and F. Filali, "Dynamic topology implementation and maintenance for the CHORIST network," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Tokyo, Japan, Sep. 2009, pp. 686–690. 9

[27] C. W. Bostian, S. F. Midkiff, T. M. Gallagher, C. J. Rieser, and T. W. Rondeau, "Rapidly Deployable Broadband Communications for Disaster Response," in *Proeedings. of he International Symposium on Advanced Radio Technologies (ISART)*, Mar. 2004, pp. 87–92. 9

[28] E. Cayirci and C. Ersoy, "Application of 3G PCS technologies to rapidly deployable mobile networks," *IEEE Networks*, vol. 16, pp. 20–27, 2002. 9

[29] J. Q. Bao and W. C. Lee, "Rapid deployment of wireless ad hoc backbone networks for public safety incident management," in *Proceedings of the Global Communications Conference (GLOBECOM)*, Nov. 2007, pp. 1217–1221. 9, 11, 12

[30] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri, "Real-Time Deployment of Multihop Relays for Range Extension," in *Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Juan, Puerto Rico, Jun. 2007, pp. 85–98. 9, 11, 12

[31] A. Wolff, S. Subik, and C. Wietfeld, "Performance Analysis of Highly Available Ad Hoc Surveillance Networks Based on Dropped Units," in *Proceedings of*

IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, May 2008, pp. 123–128. 9, 11, 12

[32] Y. Huang, "On-demand relay communication infrastructure for base station connectivity," Ph.D. dissertation, University of Illinois at Urbana-Champaign, IL, USA, 2011. 9

[33] P. Mondal, K. P. Naveen, and A. Kumar, "Optimal deployment of impromptu wireless sensor networks," in *Proceedings of the 18th National Conference on Communications (NCC)*, Feb. 2012, pp. 1–5. 9

[34] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, "Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations," in *Proceedings of the 13th IASTED International Conference on Robotics and Applications*, Würzburg, Germany, 2007, pp. 501–506. 9, 13, 14

[35] C. Q. Nguyen, B.-C. Min, E. T. Matson, A. H. Smith, J. E. Dietz, and D. Kim, "Using Mobile Robots to Establish Mobile Wireless Mesh Networks and Increase Network Throughput," *International Journal of Distributed Sensor Networks (IJDSN)*, vol. 2012, pp. 1–13, 2012. 9, 14

[36] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proceedings of the 2009 ACM symposium on Applied Computing (SAC)*, Honolulu, Hawaii, USA, Mar. 2009, pp. 1171–1175. 9, 14

[37] K.-H. Kim, K. G. Shin, and D. Niculescu, "Mobile Autonomous Router System for Dynamic (Re)formation of Wireless Relay Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1828–1841, 2013. 9

[38] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, pp. 59–67, Sep. 2008. 9

[39] D. Naudts, S. Bouckaert, J. Bergs, A. Schouttcet, C. Blondia, I. Moerman, and P. Demeester, "A Wireless Mesh Monitoring and Planning Tool for Emergency Services," in *Proceedings Workshop on End-to-End Monitoring Techniques and Services (E2EMON)*, May 2007, pp. 1–6. 11

[40] T. Aurisch and J. Tölle, "Relay Placement for Ad-hoc Networks in Crisis and Emergency Scenarios," in *Proceedings of the Information Systems and Technology Panel Symposium (IST-091)*, Bucharest, Romania, 2009. 11

[41] H. Liu, Z. Xie, J. Li, S. Lin, D. Siu, P. Hui, K. Whitehouse, and J. A. Stankovic, "An Automatic, Robust, and Efficient Multi-User Breadcrumb System for

Emergency Response Applications," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, 2013. 11, 13

[42] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Connectivity Maintenance in Mobile Wireless Networks via Constrained Mobility," *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 935–950, 2012. 15

[43] W.-W. V. Srinivasan and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of the 13th annual ACM International Conference on Mobile computing and networking (MobiCom)*, Montréal, Québec, Canada, Sep. 2007, pp. 39–50. 18

[44] L.-L. Xie and P. Kumar, "Multisource, Multidestination, Multirelay Wireless Networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3586–3595, Oct. 2007. 23

[45] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM International Symposium on Modeling, analysis and Simulation of wireless and mobile systems (MSWiM)*, Venice, Italy, Oct. 2004, pp. 126–134. 25

[46] J. del Prado Pavon and S. Chio, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Anchorage, AK, USA, May 2003, pp. 1108–1113. 25, 26

[47] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission," *Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 425–440, Mar. 2008. 26

[48] "Methods for subjective determination of transmission quality," Aug. 1996, ITU-T Recommendation P.800. 26

[49] V. C. Giruka and M. Singhal, "Hello Protocols for Ad-Hoc Networks: Overhead and Accuracy Tradeoffs," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, Taormina, Italy, Jun. 2005, pp. 354–361. 38

[50] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "What does model-driven data acquisition really achieve in wireless sensor networks?" in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Lugano, Switzerland, Mar. 2012, pp. 85–94. 38

[51] X. Li, N. Mitton, and D. Simplot-Ryl, "Mobility Prediction Based Neighborhood Discovery in Mobile Ad Hoc Networks," in *Proceedings of the 10th International IFIP TC 6 Networking Conference (NETWORKING)*, vol. 6640, Valencia, Spain, May 9-13 2011, pp. 241–253. 38

[52] A. Ghaddar, T. Razafindralambo, I. Simplot-Ryl, D. Simplot-Ryl, and S. Tawbi, "Towards Energy-Efficient Algorithm-Based Estimation in Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Hangzhou, China, Dec. 2010, pp. 39–46. 38, 41, 44, 57

[53] M. J. Luckie, A. J. McGregor, and H.-W. Braun, "Towards improving packet probing techniques," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IWM)*, San Francisco, California, USA, Nov. 1-2 2001, pp. 145–150. 38

[54] M. Hasib and J. A. Schormans, "Limitations of Passive & Active Measurement Methods in Packet Networks," in *London Communications Symposium (LCS)*, London, UK, Sep. 2003. 38

[55] T. B. Reddy, B. S. Manoj, and R. R. Rao, "On the Accuracy of Sampling Schemes for Wireless Network Characterization," in *Proceedings of the IEEE Wireless Communications & Networking Conference (WCNC)*, Las Vegas, Nevada, USA, Mar. 2008, pp. 3314–3319. 39

[56] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing.* Prentice-Hall, 1996. 40, 42

[57] C. Liu, K. Wu, and M. Tsao, "Energy efficient information collection with the ARIMA model in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, vol. 5, St. Louis, USA, Dec. 2005, pp. 5–10. 40

[58] J.-L. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized Data Aggregation in WSNs using Adaptive ARMA," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SensorComm)*, Venice, Italy, Jul. 2010, pp. 115–120. 40

[59] R. J. Hyndman, "Time Series Data Library," accessed on January 2013. [Online]. Available: http://data.is/TSDLdemo 43

[60] J. Razafimandimby, K. Miranda, D. Zorbas, and T. Razafindralambo, "Fast and Reliable Robot Deployment for Substitution Networks," in *Proceedings of the 10th ACM International Symposium on Performance Evaluation of Wireless*

Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), Barcelona, Spain, Nov. 3-8 2013, pp. 17–23. 53

[61] Wifibots, "Networked robotics," accessed on January 2013. [Online]. Available: http://www.wifibot.com/ 54

[62] R. Charette, "Cable Cut in Sydney's Central Business District," Risk Factor blog IEEE Spectrum, 2009. [Online]. Available: http://spectrum.ieee.org/riskfactor/computing/it/cable-cut-in-sydneys-central-business-district 59

[63] W. Allen, A. Martin, and A. Rangarajan, "Designing and deploying a rural ad-hoc community mesh network testbed," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, Sydney, Australia, Nov. 2005, pp. 740–743. 59

[64] Google, "Project Loon: Ballon-powered Internet for everyone," accessed on October 2013. [Online]. Available: http://www.google.com/loon/ 59

[65] A. Iera, A. Molinaro, S. Y. Paratore, G. Ruggeri, and A. Zurzolo, "Making a mesh router/gateway from a smartphone: Is that a practical solution?" *Ad Hoc Networks*, vol. 9, no. 8, pp. 1414–1429, Nov. 2011. 59

Glossary

B

Backbone A portion of a computer network that is capable of carrying the majority of traffic on the network.

Breadcrumb Small and expendable devices that act as relays.

C

Connectivity The physical, wired or wireless, and logical, i.e., protocol, connection of a node, for example, sensor or computer to a network, such as the Internet or a LAN.

Controlled mobility A direct method for physical reconfiguration based on controlled and coordinated motion of network elements or their subparts.

H

Hello packet Control packets used by some routing protocols, e.g., AODV, to exchenge control information between nodes. The messages can contain clock and timestamp information, for example.

High bandwidth A high-bandwidth connection, such as a cable modem, allows for the fast transmission of data, and significantly reduces time delays in loading.

M

Multi-hop The communication between two nodes is carried out through a number of intermediate nodes whose function is to forward information from one point to another.

Multimedia communications The transmission of a combination of at least two types of traffic data, video, and audio.

N

Next hop (X_{next}) Term used to refer to the next node to which packets should be forwarded.

P

Previous hop (X_{prev}) Term used to refer to the previous node from which packets have been forwarded.

Probe packet Artificial packets used by the active measurement techniques in order to obtain information about the network performance.

Q

Quality of communication links (Link quality) The probability that a message transmitted on the link is successfully received, i.e., the reliability of the link.

Quality of service (QoS) Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow.

R

Received signal strength indicator (RSSI) The total energy of the received signal.

Resilience The ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.

Resiliency The ability of a network to provide and maintain an acceptable level of service after one or a series of failures or disruptions.

Round trip time (RTT) The time required for a packet to travel from a specific source to a specific destination plus the time required for an acknowledgment to travel back.

S

Signal-to-noise ratio (SNR) The ratio of the desired signal energy to the total in-band noise energy.

T

Transmission rate (TxRate) The total number of physically transferred bits per second over a communication link.

Resumen

Después de un siniestro, la infraestructura de comunicación puede resultar parcial o totalmente destruida. Sin embargo, es imperativo mantener la comunicación entre los equipos de rescate y el centro de comando para coordinar las actividades de rescate. Por lo tanto, una solución temporal para establecer o reestablcer la comunicación es crucial en este tipo de escenarios. En esta tesis, nos enfocamos en una solución de comunicación llamada redes autodesplegables (o redes de sustitución). Así, primero proponemos un algoritmo de auto-despliegue para permitir que los encaminadores móviles, que componen una red de sustitución, se dispersen para cubrir un área dada. El algoritmo propuesto monitoriza las condiciones de la red para decidir si el encaminador debe moverse o no, ajustando su posición en base a la información de los nodos vecinos adquirida por medio de paquetes de prueba. Dichos paquetes de prueba permiten al algoritmo sondear el canal de comunicación y sus eventuales cambios. Si la frecuencia de transmisión de los paquetes de prueba es lo suficientemente alta, la información adquirida será precisa, sin embargo, la sobrecarga aumentará proporcionalmente el consumo de recursos de la red. Es por esto que proponemos usar un estimador basado en el modelo autorregresivo para reducir la sobrecarga en la red sin afectar el rendimiento del algoritmo de auto-despliegue. Ambas propuestas son evaludadas usando el simulador de eventos discretos NS-2 en términos del tiempo de despliegue, retardo, variabilidad en el retardo y el throughput.

Sommario

In caso di disastro, l'infrastruttura delle comunicazioni potrebbe essere parzialmente o totalmente distrutta, oppure inadeguata all'elevata intensità di traffico. Nonostante ciò, diventa necessario assicurare la connettività tra le squadre di soccorso e il centro di comando. Di conseguenza, le soluzioni di comunicazioni temponaree sono essenziali fin quando l'intera infrastruttura non viene ripristinata. L'obiettivo di questo lavoro di tesi è l'implementazione di una soluzione al problema della comunicazione detta "rete di sostituzione". Pertanto, noi proponiamo un algoritmo di self-deployment che permette ai router mobili della rete di sostituzione di distribuirsi per garantire la copertura della zona target. Il nostro algoritmo controlla le condizioni della rete per decidere se il router debba muoversi corregendo la sua posizione in funzione alle informazioni one-hop ottenute da misurazioni attive effettuate per mezzo, ad esempio, di pacchetti sonda. Questi ultimi permettono all'algoritmo di monitorare il canale e i suoi eventuali cambiamenti nel tempo. Se il rate di trasmissione dei pacchetti è sufficientemente alto, le misurazioni ottenute saranno accurate, tuttavia, l'overhead aumenterà proporzionalmente consumando più risorse di rete. Di conseguenza, noi proponiamo l'uso di dati sostitutivi, generati da un estimatore autoregressivo per ridurre l'overhead senza influenzare il nostro algoritmo di deployment. Noi dimostriamo con simulazioni l'efficienza di entrambi gli algoritmi e le loro prestazioni in termini di tempo di deployment, ritardo, jitter e throughput.

