A KEY DISTRIBUTION SCHEME TAILORED FOR MOBILE SENSOR NETWORKS

by

KEVSER KARACA

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
February 2011

A KEY DISTRIBUTION SCHEME TAILORED FOR MOBILE SENSOR NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi ....

(Thesis Supervisor)

Assoc. Prof. Dr. Ayşe Berrin Yanıkoğlu ....

Assoc. Prof. Dr. Erkay Savaş ....

Asst. Prof. Dr. Hüsnü Yenigün ....

Assoc. Prof. Dr. Özgür Erçetin ....

DATE OF APPROVAL ....

A KEY DISTRIBUTION SCHEME TAILORED FOR MOBILE SENSOR NETWORKS

Kevser Karaca
Computer Science and Engineering, MS Thesis, 2011
Thesis Supervisor: Assoc. Prof. Albert Levi

Keywords: Mobile Sensor Networks, Security, Key Distribution

Abstract

Wireless Sensor Networks, (WSN), are composed of battery-powered and resource-limited small devices called sensor nodes. WSNs are used for sensing and collecting data in the deployment area to be relayed to a Base Station (BS). In order to secure WSNs, first of all key distribution problems must be addressed. Key distribution problem is extensively studied for static WSNs, but has not been studied widely for mobile WSNs (MWSN).

In this thesis, we proposed key distribution mechanisms for MWSNs. We propose a scheme in which both sensor nodes and the BS are mobile. In our scheme, the BS works as a key distribution center as well. It continuously moves in the environment and distributes pairwise keys to neighboring sensor nodes. In this way, the network gets securely connected. We conduct simulations to analyze the performance of our proposed scheme. The results show that our scheme achieves a local connectivity value of 0.73 for half-mobile network scenario and 0.54 for fully-mobile network scenario. These values can be further improved by using multiple BSs or increasing the speed of the BS. Moreover, our scheme provides perfect resiliency; an adversary cannot compromise any additional links using the captured nodes.

We also incorporate two well-known key distribution mechanisms used for static networks into our scheme and provide a better connectivity in the early stages of the sensor network. The improvement in local connectivity, however, comes at the expense of reduced resiliency at the beginning. Nevertheless, the resiliency improves and connectivity converges to our original scheme's values in time.

MOBİL DUYARGA AĞLARI İÇİN ÖZEL GELİŞTİRİLMİŞ BİR ANAHTAR DAĞITIM YÖNTEMİ

Kevser Karaca
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2011
Tez Danışmanı: Doç. Dr. Albert Levi

Anahtar Kelimeler: Mobil Duyarga Ağları, Güvenlik, Anahtar Dağıtımı,

Özet

Kablosuz Duyarga Ağları, (KDA), duyarga düğümleri adı verilen, küçük, pille çalışan sınırlı kaynaklara sahip aygıtlardan oluşur. KDAlar veri algılamada ve toplamada kullanılır ve verileri KDAnın bulunduğu alanda yer alan Baz İstasyonuna, (Bİ), iletirler. KDAları güvenli hale getirebilmek için, öncelikle anahtar dağıtım problemleri çözülmelidir. Statik KDAlar için anahtar dağıtım problemi ile ilgili pek çok çalışma yapılmış durumdadır; ancak mobil KDAlar, (MKDA), için bu konu detaylı olarak çalışılmış değildir.

Bu tezde, MKDAlar için anahtar dağıtım mekanizmaları önerdik. Hem duyarga düğümlerinin hem de Bİnin hareketli olduğu bir model önerdik. Önerilen anahtar dağıtım şemasında Bİ, aynı zamanda bir anahtar dağıtım merkezi olarak görev yapmaktadır. Bİ sürekli ağın bulunduğu alanda hareket etmekte ve komşu duyarga düğümlerine ortak ikili anahtarlar dağıtmaktadır. Böylece ağ güvenli bir şekilde bağlı hale getirilmektedir. Önerdiğimiz şemanın performans analizini simülasyon yolu ile gerçekleştirdik. Bu simülasyonlara göre, önerilen şemanın yerel bağlantı oranı yarı-mobil ağ senaryosunda 0.54, tümüyle mobil ağ senaryosunda ise 0.73 oranlarına ulaşmaktadır. Bu oranlar Bİnin hızını artırarak ya da ağın bulunduğu alanda çok sayıda Bİ kullanarak daha da artırılabilmektedir. Ayrıca, önerilen şema düğüm ele geçirme saldırılarına karşı tam dayanıklılık göstermektedir, öyle ki; bir saldırgan ele geçirdiği düğümleri kullanarak henüz ele geçmemiş düğümler arasındaki iletişim bağlarından hiç birine zarar verememektedir.

Ek olarak, iki bilindik statik anahtar dağıtım mekanizmasını da sistemimize entegre ettik. Böylece ağın erken evrelerinde de yüksek bağlantı oranlarına ulaştık. Ancak bağlantı oranlarındaki bu artış, ağın tam dayanıklılık özelliğinden ödün vererek gerçekleşmektedir.

Acknowledgements

I would like to thank my thesis supervisor, Assoc. Prof. Albert Levi, for all his support throughout my undergraduate and graduate education and for guiding me in my studies.

I also thank Assoc. Prof. Ayşe Berrin Yanıkoğlu, Assoc. Prof. Erkay Savaş, Asst. Prof. Dr. Hüsnü Yenigün and Assoc. Prof. Özgür Erçetin for devoting their time to join my jury despite their busy schedule.

I specially thank my dearest family and friends for their constant support and love, for being there whenever I need them.

During my graduate education, I was supported by scholarships of Sabancı University and Scientific and Technological Research Council of Turkey (TÜBİTAK). I am grateful to these foundations for supporting my education.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) which consist of small battery-devices called sensor nodes have gained importance in recent years for their widespread applications [1]. The sensor nodes of the network can sense and collect data, process the data they collect or send the data to a sink node, also called Base Station. The application areas of WSNs include environmental applications, military applications, different kinds of monitoring applications, etc. The nodes in the network and the Base Station can be either static or mobile, depending on application and environmental conditions.

It is important to provide security mechanisms for WSNs like any other kind of network. However the wireless nature of communication makes network more prone to security risks and attacks. The security requirements of the network like authentication and confidentiality are done by cryptographic mechanisms of encryption and decryption. Providing WSNs with effective encryption mechanisms is a big challenge since sensor nodes are resource-limited devices which do not have high computational power and memory. This particular feature of WSNs makes them an interesting area of research and there have been many studies on security-related issues concerning WSNs.

There are two types of cryptographic mechanisms used to provide a network with necessary encryption/decryption mechanisms. They are Public Key Cryptography and Symmetric Key Cryptography. Resource-limited WSNs are not so suitable for Public Key Cryptography. Therefore generally Symmetric Key Cryptography is used in WSNs. The key distribution schemes for WSNs should provide not only end point security but also link-level security because nodes need to communicate with each other to perform certain operations like data aggregation.

The main challenge in Symmetric Key Cryptography is key distribution to the nodes. There have been many studies proposing various key distribution mechanisms for WSNs. Some of the most well-known key distribution mechanisms include Eschenauer and Gligor's Basic Scheme and Du et al.'s scheme which uses deployment information [2,3]. These probabilistic key distribution mechanisms bring a good balance of network connectivity and resilience against node capture. Other approaches for key distribution schemes include matrix-based solutions [4, 5], polynomial solutions [6] and combinatorial designs [7]. Majority of the proposed key distribution solutions are for static WSNs. In this thesis, we aim to provide a solution for key distribution problem for mobile WSNs.

1.1 Our Motivation and Contribution of the Thesis

The concept of mobile wireless sensor networks, which has emerged later than static WSNs, refers to networks that have a mobile sink and/or sensor nodes [8]. There has been some research on mobile WSNs regarding their differences from static WSNs, their possible advantages or disadvantages over them. The dynamic topology of the network due to mobile nature of sink and/or nodes, more challenging routing problems, possible renewal of energy at gateway sink and efficient energy use is some of their differences from static WSNs, as mentioned in [8]. Other issues of mobile WSNs like coverage problem [9], deployment of the network [10, 11] have also been addressed in several papers. However, security issues, in particular key distribution mechanisms have not been studied much, especially compared to static WSNs.

In this thesis, we study key distribution problem from mobility perspective and propose a key distribution scheme for mobile Wireless Sensor Networks. We first discuss the existing key distribution schemes, and then analyze two of the schemes' performance in mobile WSNs. Our analysis show that location information based solutions are greatly affected by mobility and they perform very badly under mobile conditions. Our scheme, on the other hand, is especially designed for mobile WSNs. We use Base Station as a mobile key distribution center which provides the nodes with pairwise keys of their neighbors as they meet with Base Station. We conducted simulations for various cases for our scheme and calculate performance metrics like local connectivity, global connectivity and resilience. The simulations show that our scheme achieves a local connectivity between 0.45 and 0.74 depending on the parameters used in different cases. Moreover global connectivity values are close to 1. Our scheme also has perfect resilience property such that an adversary cannot compromise any addition links using the nodes he/she captured previously. We also propose modifications to our scheme to introduce higher connectivity directly from the start of deployment. We incorporate other schemes, Basic Scheme and Du's Scheme into our scheme. These modified schemes also achieve results similar to our original scheme.

1.2 Organization of the Thesis

The rest of the thesis is organized as follows: In Chapter 2, background information is given about Wireless Sensor Networks, cryptographic overview and security requirements of the WSNs. This chapter also includes background information about previously proposed key distribution schemes and mobility models used for mobile networks. In Chapter 3, we introduce our scheme and show our scheme's performance for various cases in terms of local connectivity, global connectivity and resilience. Chapter 4 explains how other schemes, namely Basic Scheme and Du's Scheme, are incorporated into our scheme and shows the performance of modified schemes. Finally Chapter 5 concludes the thesis.

Chapter 2

Background

2.1 Wireless Sensor Networks (WSNs)

Wireless sensor networks (WSNs), are composed of large number of sensor nodes which are small, battery-powered devices [1]. There are different typse of Wireless Sensor Networks. A WSN can be hierarchical or which consists of Base Stations, cluster heads and sensor nodes. In hierarchical WSNs sensor nodes generally communicate with cluster head rather than communicating with other sensor nodes. Cluster heads also have some hierarchy depending on the application and the data is processed in a hierarchical way and sent to Base Stations. A WSN can also be distributed with no fixed infrastructure and with unknown network topology prior to deployment. There are still Base Stations in distributed WSNs as well, but communication is not done in a hierarchical way. In this thesis, we take distributed WSNs into consideration. A WSN can also be either static in which nodes and Base Station are immobile, or mobile in which node and/or Base Station are mobile.

WSNs have a wide range of application areas, like military applications, health applications, environmental applications, etc. They collect data and transmit them using integrated radio communication interface. They have some differences from the ad hoc networks like their number being much higher in the network, sensor nodes being densely deployed, being more prone to failures, and having a more dynamic topology. In addition to these, sensor nodes are low-cost devices; they operate on low power and they have limited memory. This brings a big constraint on the solutions offered for sensor networks, since they must be energy efficient. Therefore, many of the solutions offered for ad hoc networks are not suitable for sensor networks.

2.2 Security Requirements

Sensor networks can be deployed in various areas, some of which may be hostile environments. It is important to provide necessary mechanisms to provide security in the network. Some of the security needs of wireless sensor networks are listed below [12, 13]:

- *Confidentiality* is the basic security service to keep the secrecy of the important data and to allow only the authorized party to access information. The standard way to provide confidentiality is to use encryption with a secret key.
- *Authentication* means that a receiver should be able to verify that the data is really sent by the real sender. To ensure authentication, the sender should provide a cryptographic code of the message using a key and the receiver should be able verify the code and identify the sender.
- *Integrity* must be kept to ensure that the transmitted data does not get modified by unauthorized people during transmission.
- *Data freshness* means that the data is recent and no old messages have been replayed.
- *Availability* means that the WSN is able to provide service whenever it is required.
- *Secure localization* refers to methods that give the network the ability to accurately locate each sensor in the network.

2.3 Cryptographic Overview

Cryptographic protocols are used to ensure the security requirements mentioned above are met. However due to wireless nature of communication and limited capabilities of sensor nodes, certain limitations apply to the use of these cryptographic protocols. In general, there are two approaches to provide necessary cryptographic protocols; namely symmetric key cryptography and asymmetric key cryptography.

In symmetric key cryptography, a single key is used for both encryption and decryption. That means the same key should be supplied to all the authorized parties to enable them encrypt and decrypt the messages sent by their corresponding partners in communication. Distribution of the single key to multiple entities is the main challenge for symmetric key encryption. Encryption and Decryption mechanism for symmetric key cryptography is shown in Figure 2.1.

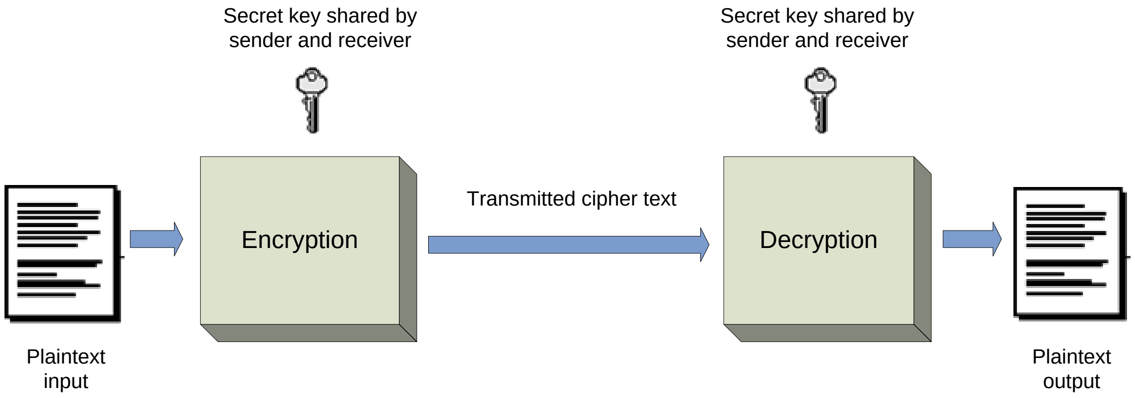

Secret key shared by sender and receiver
Secret key shared by sender and receiver
Encryption
Transmitted cipher text
Decryption
Plaintext input
Plaintext output

Figure 2.1 Encryption and decryption mechanism for Symmetric Key Cryptography

In asymmetric key cryptography, which is also known as public key cryptography, two separate keys are used. One of the keys is public and known to other entities in the communication, while the other key is kept private and known only by its owner. Encryption of a message is done by the public key of the receiver and decryption of a message is done by using the private key of the receiver. By this mechanism, no one but the rightful receiver of the message can decrypt the message. Figure 2.2 describes the encryption and decryption mechanism for asymmetric key cryptography.

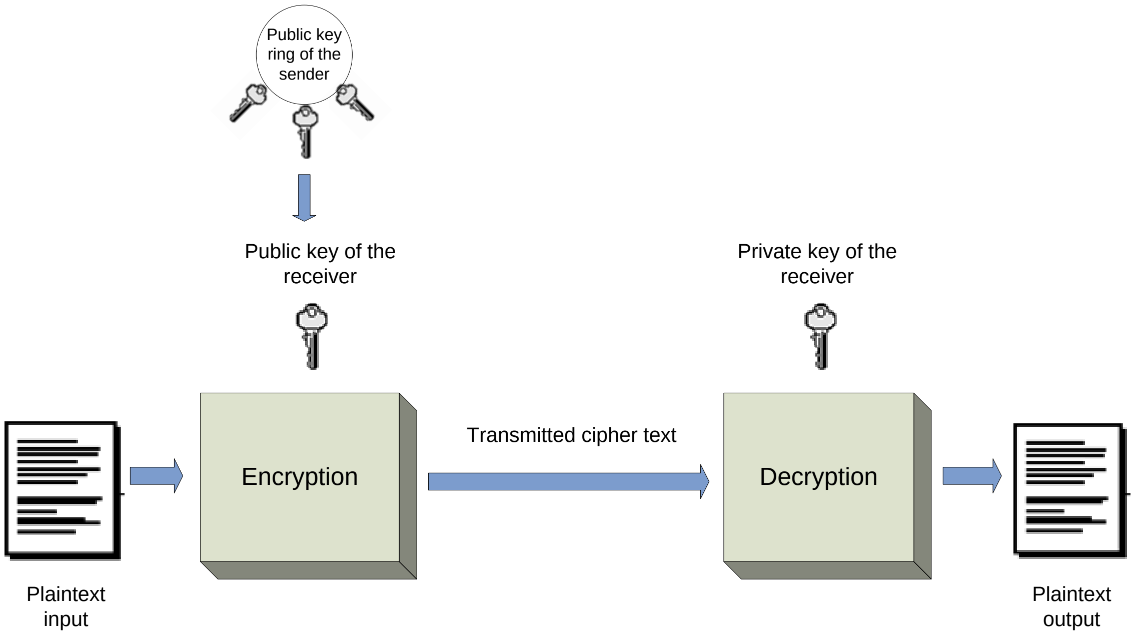

Public key ring of the sender
Public key of the receiver
Private key of the receiver
Encryption
Transmitted cipher text
Decryption
Plaintext input
Plaintext output

Figure 2.2 Encryption and decryption mechanism for Asymmetric Key Cryptography

Asymmetric key cryptography has some advantages over the symmetric key cryptography. One advantage is the relative easiness of asymmetric key distribution as compared to symmetric key distribution. Another advantage is that private key ownership implies the identity of one unique entity; however symmetric key means the existence of same key in different entities. This makes authentication a more challenging job for symmetric key encryption.

Asymmetric key cryptography is widely used in computer networks because of its advantages mentioned above. However, they require more amount of energy and computational power compared to symmetric key cryptography and this makes them unsuitable for resource limited networks like WSNs. There are some research that investigate ways to implement asymmetric key cryptography in WSNs [14, 15, 16], yet these proposals still need higher energy and computational power, thus, they seem infeasible for WSNs.

The above-mentioned disadvantage of asymmetric key cryptography made the researchers investigate several ways to deal with the key distribution problem in symmetric key cryptography. There are several approaches and many suggestions to solve this problem in WSNs. These approaches to the keys distribution problem and related work in the field is explained in the next section.

2.4 Literature Survey of Key Distribution in WSNs

There are lots of works on key distribution on WSNs in the literature. There are many surveys on the issue as well. The papers by Çamtepe and Yener [17], Zhang and Varadharajan [18], Zhou et al. [13], Lee et al. [19] and Xiao et al. [20], M. A Simplício Jr. [21] et al. provide good surveys on the general key distribution problem and taxonomy of the works proposed for WSNs. Main approaches used for key distribution problems and well-known works in the area are explained below.

One straightforward solution to key distribution problem in WSNs is using a single master key for all the nodes in the network. In this method, all nodes are given the same key and communication between nodes is done by this key. The advantage of the method is the perfect connectivity it brings to the network. However this also is the biggest weakness of the system. Capture of one node is enough to get access to all the communication of the network. Therefore it has the worst performance in terms of resilience against node capture attacks. There are some proposals using this idea of single key in the network. One of them is Broadcast Session Key Negotiation Protocol (BROSK) [22]. This protocol uses a master key with random nonces between nodes. A session key between nodes is created by applying a pseudo-random function to the master key concatenated with the session keys. Another such protocol is Symmetric-Key Key Establishment protocol which uses exchange of randomly created challenges of a predetermined length and master key to compute a shared secret, which is subsequently used to create session key for the communication [23]. Another protocol using this idea is Loop-Based Key Management Scheme [24]. This scheme uses master key together with individual keys and key IDs to create session keys.

Another straightforward solution to the key distribution problem is to load each node pair-wise keys between that node and of every other node. In this scheme if a network consists of *n* nodes, each node in the network carries *n*-1 pair-wise keys in its memory. The scheme has perfect resilience since capture of a node does not cause the compromise of non-captured nodes' links. The problem with this scheme is the memory overhead especially if the network is large. Since sensor nodes are limited-memory devices, this scheme is not suitable for them.

Other than the above-mentioned straightforward but unsuitable key distribution schemes, there are works that try to solve the key distribution problem with different methods. Among these methods, probabilistic solutions are one of the most famous and mostly used methods. These solutions, which will be explained below, bring a tradeoff between network connectivity and network security compared to other approaches like matrix-based solutions and polynomial-based solutions. Probabilistic solutions do not guarantee perfect connectivity; they achieve connectivity with certain probability; however they offer better security in terms of the number of compromised links in case an attacker captures some of the nodes in the network.

The initial idea to introduce a probabilistic approach to key distribution problem is proposed by Eschenauer and Gligor [2]. This scheme which is referred to as Basic Scheme, introduces the concept of *key pool* and *key chain*. There is a global key pool for the network produced beforehand. The scheme has three important phases; key pre-distribution, shared-key discovery and path-key establishment phases.

- *Key Predistribution Phase:* Prior to deployment, each sensor is loaded with a set of keys randomly selected from the global key pool. This set of keys is called key chain of a node. A node keeps this set of nodes in its memory and uses these keys in communication later. Please note that it is possible that two nodes have the same key in their key-chains. This property is actually essential for communication in probabilistic approaches. After key pre-distribution nodes are deployed uniformly to the environment.
- *Shared-key Discovery Phase:* After deployment, shared-key discovery phase begins. At this phase, nodes try to find their neighbors and figure out whether they have common keys or not. Two nodes can communicate with each other if they have at least one common key in their memory. Such secure links between neighboring nodes are called direct links.
- *Path-key Establishment Phase:* There might be cases where two nodes do not have direct links between them. When such a case occurs, path-key establishment starts. At this phase, nodes try to find a path between them

through their neighbors with whom they have direct links, so that they can reach each other via secure links.

There have been many schemes that use the idea by Eschenauer and Gligor and make modifications to the Basic Scheme. One such scheme is the q-Composite scheme proposed by Chan et al. [25]. In the Basic Scheme, one shared key is enough for two nodes to establish a direct link between them. In q-composite scheme, however, two nodes need to have at least q shared keys where $q > 1$ to establish a direct link between them. This restriction is done to increase the network resilience. Another modification to the Basic Scheme includes Session Key Scheme which creates session keys for interaction of nodes using the key found at shared-key discovery phase of Basic Scheme [26], Hashed Random Key Predistribution which improves the resilience [27], Key Redistribution Scheme which proposes another phase instead of the path-key establishment phase [28], and the Pairwise Key Establishment Protocol that decreases the communication overhead of path key establishment phase [29]. Basic Scheme and other schemes mentioned above use the predistributed keys for the full lifetime of the network and this can create security vulnerabilities. There are some other schemes that address multiple deployment scenarios like Robust Key Pre-distribution (RoK) scheme [30] and Random Generation Material Scheme [31].

One other approach to the key distribution problem is matrix-based solutions. The original idea of the matrix-based solution is by Blom [4]. It is a multipurpose deterministic key pre-distribution scheme. The basic idea is that all possible keys in a network of size N can be represented by an NxN matrix. Every node can calculate its pair-wise key with another node provided that it carries $\lambda+1$ keys where $\lambda << N$. The scheme has λ-secure property which means an adversary cannot compromise any links if it has captured less than λ nodes, however it can compromise all the links once it has captured λ nodes.

Du et al. [5] proposed a scheme called Multiple Space Key Predistribution Scheme which uses Blom's scheme and Basic Scheme to improve the resilience of Blom's scheme without increasing λ. Instead of using single key space, it uses multiple key spaces. In key predistribution phase, τ different key spaces are picked randomly for each node from a key space pool. In the shared key discovery phase, two nodes can generate a pair-wise key and communicate with each other if they share key material from the same key space. Du et al.'s scheme increases the resiliency of Blom's scheme while turning the scheme to a probabilistic key distribution scheme. There is a tradeoff between resiliency and connectivity and it has additional memory overhead, since this scheme requires multiple key spaces to be kept in nodes' memory. Other matrix-based solutions include the work by Lee and Stinson [32] which improves the scalability of the scheme, the work by Chien et al.[33], which introduces a temporary master key to improve the resilience of Blom's scheme.

Polynomial-based schemes represent another approach used for key distribution in WSNs. Blundo's scheme is one of the best known schemes among these proposals [6]. In this scheme, a randomly-generated λ-degree polynomial is used which satisfies the rule $f(x, y) = f(y, x)$. At the key predistribution phase each node i receives a polynomial share $f(i, y)$. At the key establishment phase the nodes i and j exchange their IDs and calculate the key $K_{ij} = f(i, j) = f(j, i)$. Like Blom's scheme, Blundo's scheme also has λ-secure property and perfect connectivity. The work by Liu and Ning [34] which takes the initial idea of Blundo and uses it in a polynomial pool-based key predistribution scheme, is another scheme that uses this approach and combines it with pool based key distribution schemes to strengthen resilience and scalability.

Another approach to key distribution problem is combinatorial design. This approach assumes that the distribution of nodes can be modeled by combinatorial design techniques. Therefore, before distribution each node is loaded with keys that are carefully chosen in a deterministic and optimized manner. Some of these techniques require dense networks to function properly, since the proximity of nodes are important for connectivity, while there are some works that can function in sparse networks as well. The schemes proposed by Çamtepe and Yener [7], the scheme of Lee and Stinson [32] and the scheme proposed by Gupta and Kuli [35] are some of the works that try to solve key distribution problem using this approach.

There are also some schemes that try making use location information of the nodes along with the key distribution approaches explained above. One of the best known examples of this method is Du et al.'s scheme [3]. The idea for this scheme is that since nodes need to communicate with their neighbors in the first place, they do not need to share keys with nodes that are geographically far from themselves. Instead, they should share keys with their neighbors to provide a better connectivity for the network.

Similar to Basic Scheme, Du et al.'s Scheme also has three phases, namely key predistribution phase, shared-key discovery phase and path-key establishment phase.

- *Key Predistribution Phase:* In this scheme, the global key pool is divided into smaller groups of key pools. The deployment area is also divided into zones and each key pool is associated with a zone. Figure 2.3, shows an example of zone division and relation of the key predistribution for each zone.

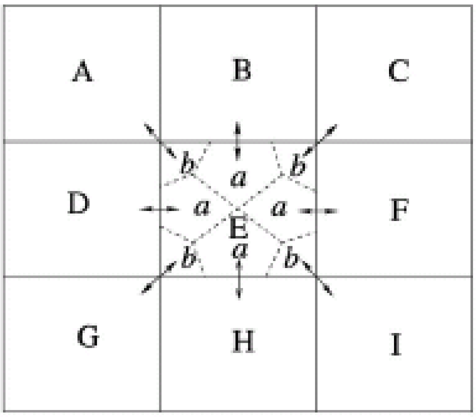
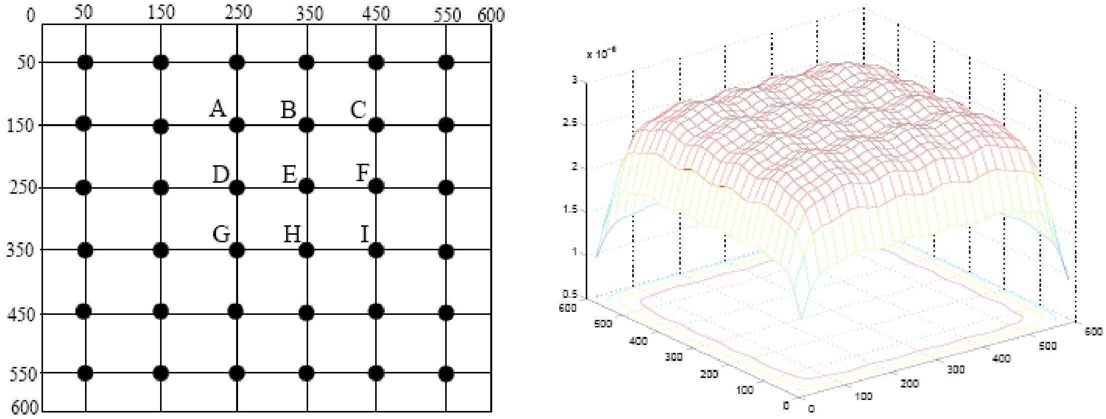


Figure 2.3 Shared keys between neighboring nodes for Du et al.'s Scheme [3]

The nodes that will be deployed to a certain zone are loaded with keys which are selected randomly from the key pool associated with that zone. As it can be seen in Figure 2.3, key pool of a particular zone gets keys from its neighboring zones' key pools as well. This way, nodes of neighboring nodes can share a common key and communicate with each other even if they belong to different zones. After key predistribution phase, nodes are deployed to the each zone using Gaussian distribution. The center of each zone is the deployment point for each Gaussian distribution. Figure 2.4 illustrates the distribution of the nodes.



(a) Deployment points (each dot represents a deployment point).

(b) Deployment distribution on the entire region using the deployment strategy modeled by (a).

Figure 2.4 Node deployment for Du et al.'s Scheme [3]

- *Shared-key Discovery Phase:* After deployment, nodes try to find out whether they have a common key with their neighbors or not. If two neighboring nodes have a common key, then there exists a direct link between these two nodes.
- *Path*-key Establishment Phase: If a node does not have a common key with another node, it uses its neighboring nodes with which it has a direct link, and tries to find a path to reach the other node. If they can find such a path, then they have a secure link.

This scheme which will be referred to as Du's Scheme in this thesis achieves higher connectivity than the Basic Scheme using the same amount of keys per node. The reason for that is the efficient use of the key pool by making use of the location information.

Other schemes that use location information together with probabilistic approach include Liu and Ning's scheme which uses Blundo's scheme and location information together to achieve better resilience and connectivity [36, 37], Zone Based Robust Key Distribution, Zo-Rok which combines group based deployment scheme with RoK and achieves better resilience of the network [38], Yu and Guang's scheme which combines location information with matrix-based approach [39]. The schemes proposed in [40, 41] also make use of location information in their proposed solutions.

The schemes explained above, especially schemes which use location information are mostly directed at static WSNs. Compared to the schemes proposed for static WSNs there are few works done about key distribution in mobile WSNs. Such schemes which take mobility into consideration to some degree are explained below.

One of the earliest works that discusses the key distribution problem from the mobility point of view is by Čapkun et al. [42]. Their proposal is for ad hoc networks, not particularly for WSNs, however the idea they present is important for WSNs as well. The fundamental argument of their work is that, mobility is not a problem for security in mobile networks; on the contrary it provides a medium where security associations like authentication and key distribution can be established. The idea is to exchange mutual credentials between nodes when they are in the close range of each other. Public key cryptography is used in their proposal. There are two cases considered; in the first one, there is no central authority, the network is fully self-organized; in the second one, there is an offline authority which provides the authorization to the nodes when they first join the network. In fully self-organized case, they assume a secure side-channel (by physical contact or infrared) on which no adversary can modify the messages transmitted over the channel; however confidentiality of the messages is not required. Every node can generate cryptographic keys, check signatures, etc. When two nodes come in to the close range of each other, they decide whether they will trust each other and establish a security association or not. If they will establish a security association, then they activate the secure side-channel and exchange related material over the channel. In the presence of a central authority, each node is given their certificate that is signed by the central authority and the public key of the central authority just before they enter the network. The authors later extended their work by including establishment of security associations with symmetric key cryptography [43]. In this case, the exchange of key material is done over secure side-channel which provides both integrity and confidentiality.

Most of the works done on key distribution mechanisms in WSNs assume static WSNs for their propositions. There are few that pay attention to mobile WSNs. One of the papers that take mobility into consideration to a degree is by Zhou et al. [44]. In their paper they propose a group-based key predistribution scheme. In the scheme sensor nodes are deployed to the area in groups. The deployment model they propose is a flexible one. A group can land on any part of the area, but still nodes of the same group are neighbors with high probability. Each node in one group shares pair-wise keys with each member of its own group. For intra-group key predistribution, they already have unique pair-wise keys. For inter-group key predistribution, key establishment is done with the help of *agent* nodes. A node s_i from group G_u, represented as $<G_u, s_i>$ is called an *agent* for G_v in G_u , if $<G_u, s_i>$ is associated, that is; it shares a *preloaded* unique pair-wise key with some $<G_v, s_j>$ in G_v. Each group is required to be associated with every other group since any group can be their neighboring group. When two nodes from different groups want to establish a pair-wise key with each other, key exchange is done through agent nodes. The authors point out that the scheme they propose is suitable for both static sensor nodes and for nodes that move in a swarm fashion [45]. However the scheme is not suitable for other mobility models where the nodes move independently of each other, since finding agent nodes of its own group in a reachable distance would be much harder in such a case.

Another work which also tolerates mobility to a mild level is done by Ünlü and Levi [46]. In this work, there are two kinds of nodes in the network; regular nodes and agent nodes. The number of agent nodes is much smaller than the number of regular nodes and they are more capable as well; they have more memory and power. The deployment of the nodes is done in groups to the grids in the area called *zones*. Regular nodes of the same group can initially only share keys with nodes of their own groups. Intra-zone key predistribution in the proposed scheme is done in a similar way to the method proposed in [47]. Inter-zone key predistribution is done only between agent nodes. Random pair-wise keys are distributed to the agent nodes. An agent node shares a unique random pair-wise key with every other neighboring agent nodes, that is; the agent nodes from the eight zones surrounding its own zone/group. An agent node also gets keys from intra-zone key predistribution method to communicate with its own group members. The paper also provides intra-zone and inter-zone path key establishment methods to be used after deployment. If two regular nodes from different groups want to communicate with each other, they establish a key using the agent nodes in their groups. If a node is drifted to a neighboring zone, it can still establish keys with the nodes through intra-zone path key establishment method if it encounters a node from its own zone, or inter-zone path key establishment method if it encounters a node from the neighboring zone, provided that it has some neighbors around which it shares a key with. This method they propose works only if the node drifts one zone away. For highly mobile networks the method does not work as authors also point out.

Key distribution scheme proposed by Dong et al. works for mobile WSNs as well as static WSNs [48]. In this scheme, there are regular sensor nodes and *assisting nodes*. Assisting nodes are only responsible for key establishment and management; they do not perform any kind of sensing and forwarding job. Each regular node has an ID and unique pair-wise key which it shares with the base station. Every assisting node *i* gets preloaded by the hash $H(K_u||i)$, for every regular sensor node *u* which shares a pair-wise key K_u , with the base station. When a node *u* wants to establish a key with a node *v*, it discovers the assisting nodes in its neighborhood and sends a message with its ID to the assisting nodes. Every assisting node *i* that got the message in the neighborhood generates a random key and encrypts the key with both $H(K_u||i)$ and $H(K_v||i)$, concatenates them and sends them to node *u*. Nodes *u* and v run a protocol similar to Needham-Shroeder Symmetric Key Protocol [49] to decrypt the key generated by the assisting node. Final key is produced by XORing all the random keys sent by the assisting nodes. If no assisting node is found in the immediate neighborhood, then the node searches for assisting nodes that are at certain amount of hops away and runs the same protocol for them. This scheme handles both static WSNs and highly mobile WSNs. The disadvantage of the scheme however, is the ratio of assisting nodes to the network size necessary to achieve a high connected network. For the connectivity to be 90%, the ratio of assisting nodes to the network size must be about 1/10 which means for a network with 10,000 nodes, there has to be 1,000 assisting nodes. This might be undesirable since assisting nodes do not perform any work that regular nodes do. This ratio can be reduced, however this time the number of hops a node needs to search to find an assisting node increase, which means more broadcast messages, thus an increase in communication cost.

2.5 Mobility Models

There are various mobility models for mobile networks in the literature. Some of the models are for independent nodes moving on their own while some are for group mobility. Some of the important mobility models used in mobile networks are explained below.

One of the well-known mobility models is the *Random Walk* Mobility Model [50]. It is a basic mobility model that uses random direction and speed. In this model, a node moves from its position to another by choosing a random direction between [0, 2 п] and a random speed between [*speedmin, speedmax*]. *speedmin and speedmax* are predefined values. The node moves in that direction with that speed for a constant time *t* or a constant distance *d*. Once this movement is completed, the node calculates a new direction and speed and repeats the same procedure. If it reaches the end of the simulation area, it bounces back, meaning it gets a new direction determined by the direction it came to the boundary and moves away. There are variations to this model like 1-*D*, 2-*D*, 3-*D* and *d-D* walks. There can also be simplifications like choosing a uniform speed for all the nodes and such. It is also a memoryless model, which means it has no memory of its past locations and speeds. Figure 2.5 shows the traveling pattern of a single node using Random Walk Mobility Model.

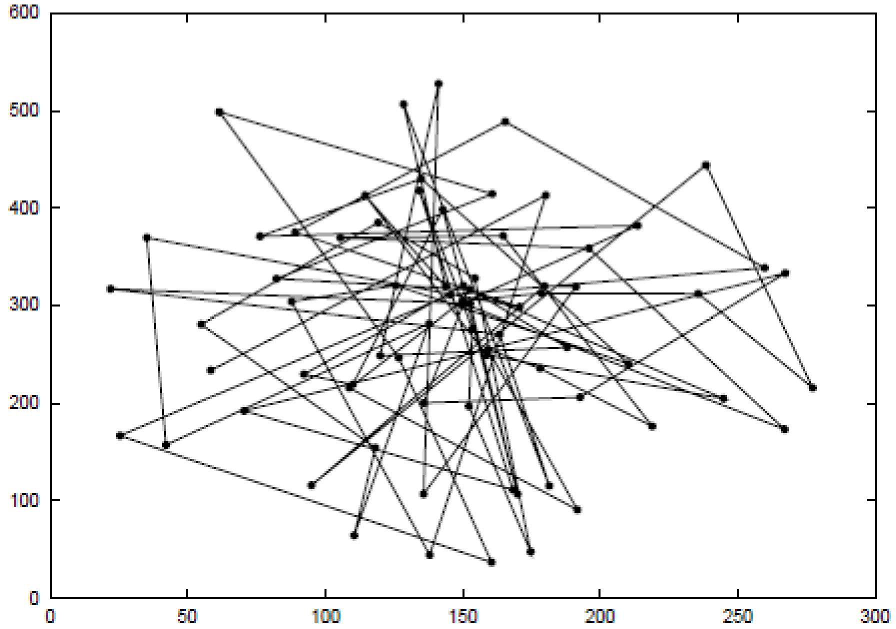


Figure 2.5 Traveling pattern of a single node using Random Walk Mobility Model [50]

Another important mobility model is the *Random Waypoint* Mobility Model. This model introduces pause times. A node starts with staying for a certain amount of time at its initial location. After this, it randomly chooses a destination and speed between predefined [*speedmin, speedmax*], and starts moving towards its destination. Once it reaches its destination, it again pauses for a specified time and starts moving in the same pattern after that pause time. This model is also a widely used mobility models. It can also be simplified by omitting the pause time from the model. Figure 2.6 shows the traveling pattern of a single node using Random Waypoint Mobility Model.

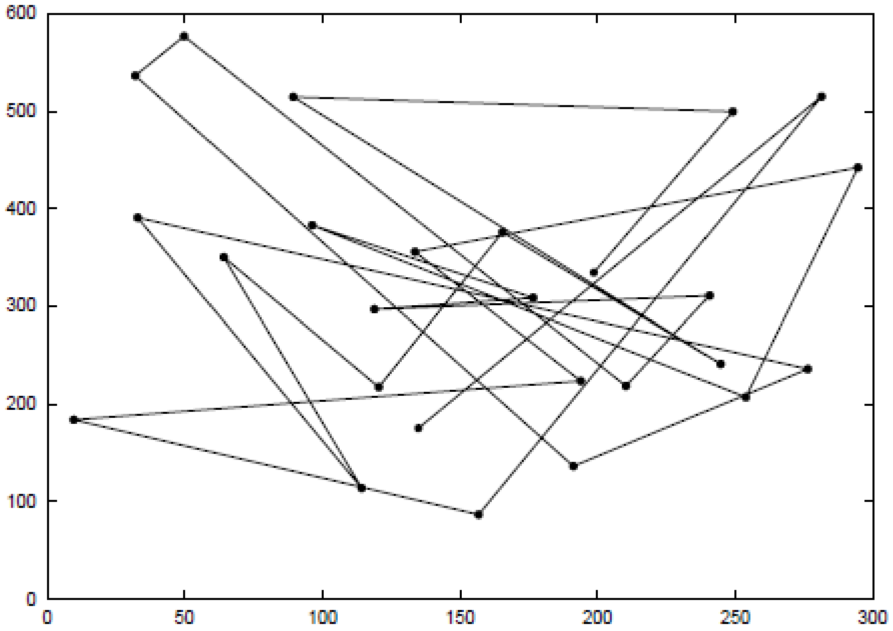


Figure 2.6 Traveling pattern of a single node using Random Waypoint Mobility Model [50]

Other mobility models for independently moving nodes include the Random Direction Model, which forces the nodes to travel until the border of the simulation area is reached, the Boundless Simulation Area Mobility Model, in which the nodes do not bounce back but continue to move and reappear on the opposite side of the area once a border is reached, the Garkus-Markov Mobility Model, City Section Mobility Model etc. There are also mobility models for group-based mobility like Nomadic Mobility Model, Pursue Mobility Model, Reference Point Group Mobility Model, Swarm Mobility Model etc [50].

Chapter 3

A Key Distribution Scheme for Mobile Wireless Sensor Networks

3.1 Effects of Mobility on Basic Scheme and Du's Scheme

Basic Scheme by Eshenauer and Gligor [2] and Du's Scheme [3] are two well-known key distribution schemes for WSNs. These schemes use a probabilistic approach to achieve high connectivity and security for the network. Among these solutions Du's Scheme uses location information of the nodes to achieve a better connectivity than Basic Scheme. One of the most important performance metrics of key distribution schemes is local connectivity. Local connectivity is the probability of any two neighboring nodes sharing a key [3]. The local connectivity ratio for static WSNs achieved by Basic Scheme and Du's Scheme with respect to the x-axis value *m*, which is the number of keys each node keeps in its memory is shown in Figure 3.1

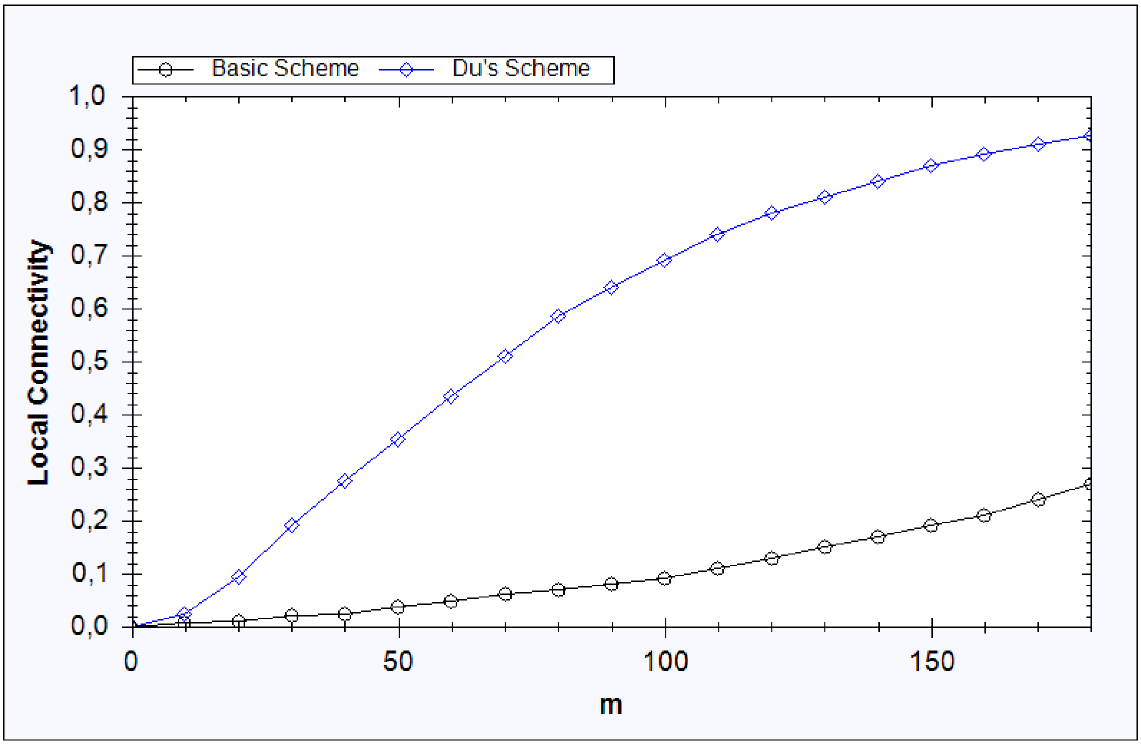


Figure 3.1 Local connectivity versus *m* for Basic Scheme and Du's Scheme for static WSNs

As seen in Figure 3.1, Du's Scheme achieves a much higher local connectivity compared to Basic Scheme. This is achieved by making use of the deployment locations of the nodes which was explained in Section 2.4.

As explained in Section 2.4 these solutions do not take mobility into consideration. In this part, the effect of mobility on these two key distribution scheme is explained.

3.1.2 Effects of Mobility on Basic Scheme

To observe the effects of mobility on Basic Scheme we conduct time-dependant simulations for two cases of node behavior. We fix *m*, the number of keys a node has to 100 to focus on the time dimension. Figure 3.2 shows mobile nodes' local connectivity values versus time using random walk mobility model.

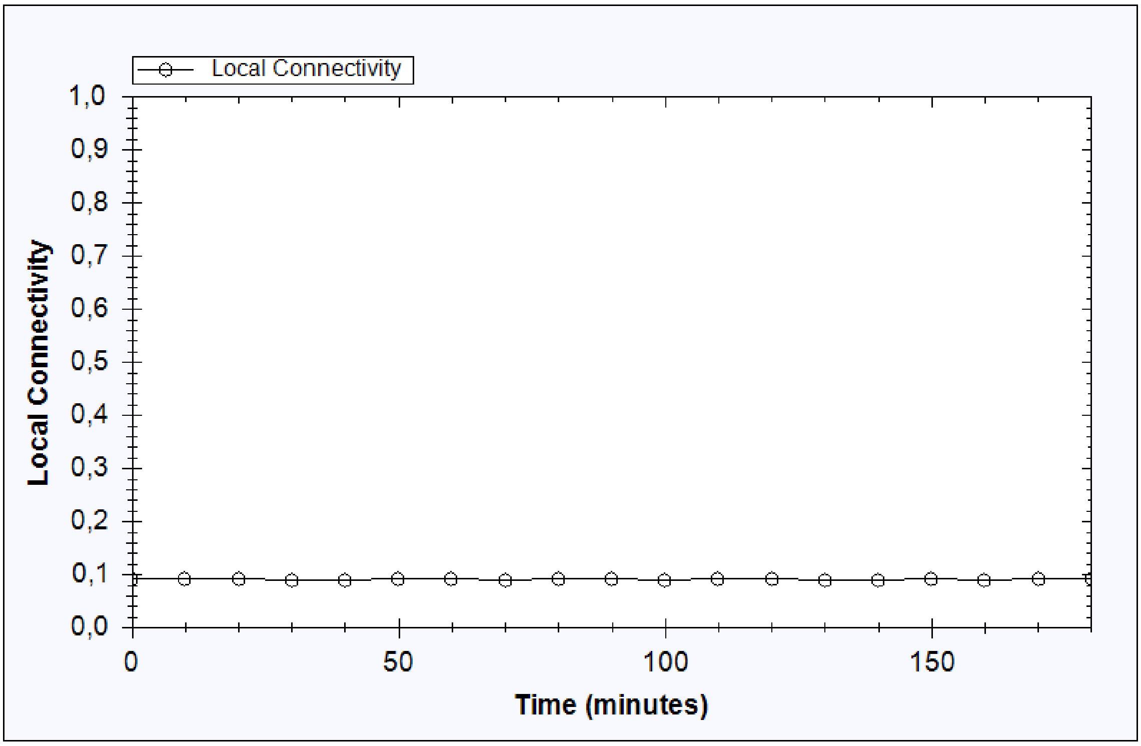


Figure 3.2 Local Connectivity versus time for Basic Scheme using Random Walk Mobility Model for *m*=100

If we compare the connectivity results from this graph with the actual values shown in Figure 3.1, it can be seen that the connectivity values for *m*=100 in this graph have the same values as it has in the original graph. We can see that local connectivity values do not change over time. This indicates that mobility of the nodes does not have any effect on the Basic Scheme. This result is actually to be expected, because in Basic Scheme, the keys put into each node is chosen at random without any regard to their deployment area, which means that the probability of two nodes next to each other sharing a key and probability of two nodes at two distant locations sharing a key would be the same. This is also the reason why mobility does not affect Basic Scheme's local connectivity values, since the probability of any two nodes sharing a key does not change with regards to their geographic location.

3.1.3 Effects of Mobility on Du's Scheme

To observe the effect of mobility on Du's Scheme, we first conduct a time dependent scenario. All the system requirements, such as the number of the nodes, the number of the zones, and the area of each zone are the same with the original values set in [3]. In this scenario *m*, the number of keys put into each node is kept constant and set to 100. The x-axis represents the time during which the nodes move in the environment is changed. At each step the nodes are left to move in the environment according to Random Walk Mobility Model for the given amount of time and after their movements local connectivity is calculated. The results of the simulations can be seen in Figure 3.3.

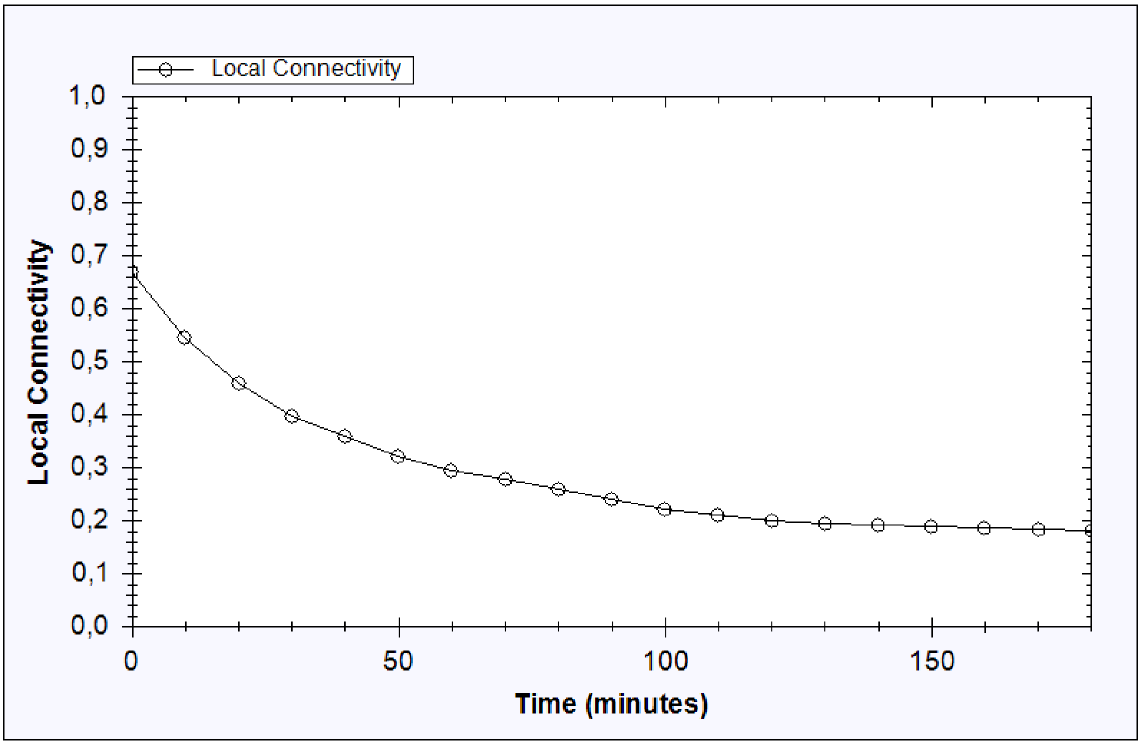


Figure 3.3 Local Connectivity versus time for Du's Scheme using Random Walk Mobility Model for *m*=100

As it can be seen from Figure 3.3, mobility has an important effect on local connectivity in this scheme. Please also note that connectivity reaches a constant value as time passes in both of the figures. This value is reached around *time*=180, and local connectivity value is around 0.18.

Additionally to observe difference between mobile and static cases, we also conduct *m*-dependant simulations. In this scenario the number of nodes put into each node changes and time is kept constant. The value for time is 180 minutes. During simulations, first local connectivity is calculated before any of the nodes start moving. This is referred as "not-mobile" in the figures shown below. After that, nodes move around in the neighborhood for the given fixed amount of time. After their movement, local connectivity is once again calculated. This is referred as "mobile" in the figures. The simulation result is seen in Figure 3.4.

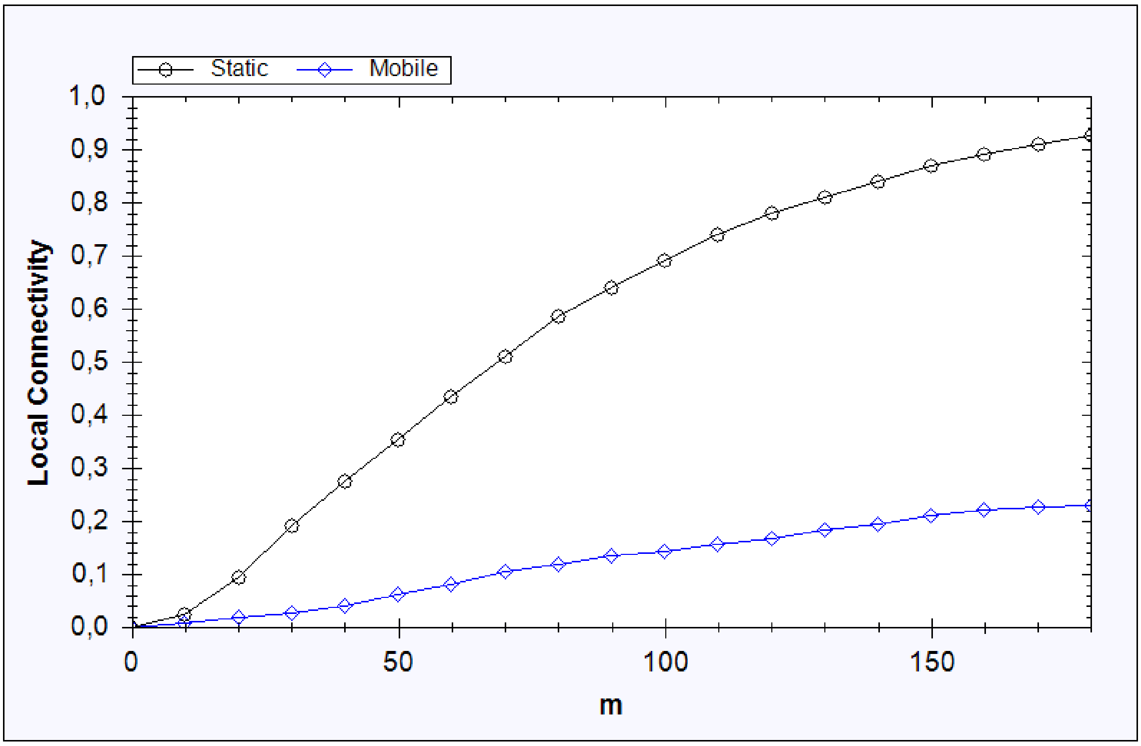


Figure 3.4 Local Connectivity for Du's Scheme for static and mobile cases with Random Walk Mobility Model where *time*=180 minutes

Looking at Figure 3.4, it is easily seen that there is a big difference between mobile and not-mobile cases in terms of local connectivity. This shows that mobility has a considerable impact on Du's Scheme. This result is what we expected, because keys are shared between nodes that are close to each other and distant nodes do not share keys; which means the locations of the nodes are extremely important for this scheme to keep connectivity high. When nodes start moving in the environment, they go to different locations nodes which do not have any common keys with each other become neighbors which results in a decrease in local connectivity when the nodes are mobile.

3.2 Our Scheme: A Key Distribution Scheme Tailored for Mobile Sensor Networks

We propose a key distribution scheme for mobile Wireless Sensor Networks. In our scheme, we propose that both nodes and the Base Station, (BS), are mobile. Mobile base stations are also used in works like [8, 51, 52]. The main idea of our scheme is to have BS operate as a key distribution center throughout the life of the sensor network. The Base Station in our scheme is mobile and tamper-proof. In our scheme, prior to deployment, nodes are not preloaded with any keys. After they are deployed to the area, BS starts to move in the simulation area and distribute pairwise keys to neighboring nodes it meets along the way.

In our scheme we use Random Walk Mobility Model for the movement of the nodes in order to have independently moving entities rather than group movement. In our version of Random Walk Mobility Model, a node randomly selects a direction between $[0, 2\pi]$, and a speed between [*speedmin*, *speedmax*]. The node moves in that direction for 1 minute and chooses a new direction and speed without waiting at that point and continues its movement. If it meets to boundaries of the simulation area, it bounces back.

For the movement of BS, we use a deterministic approach to ensure that BS scans the whole area and meets with possibly all the nodes. In our mobility model for BS, it starts moving from one bottom-corner of the simulation area, goes to the opposite edge horizontally. After it gets very close to the boundary, it starts moving vertically for a very short distance, and then starts moving horizontally again. When it scans the whole simulation area, it diagonally goes back to a bottom-corner and starts its movement from there again. Figure 3.5 shows an illustration of the movement of BS. The point denoted as *S* in the figure is the start point of the Base Station, after it completes its one round in the simulation area as shown in the figure, it starts its next round in the same way and keeps moving throughout the whole life of the network.

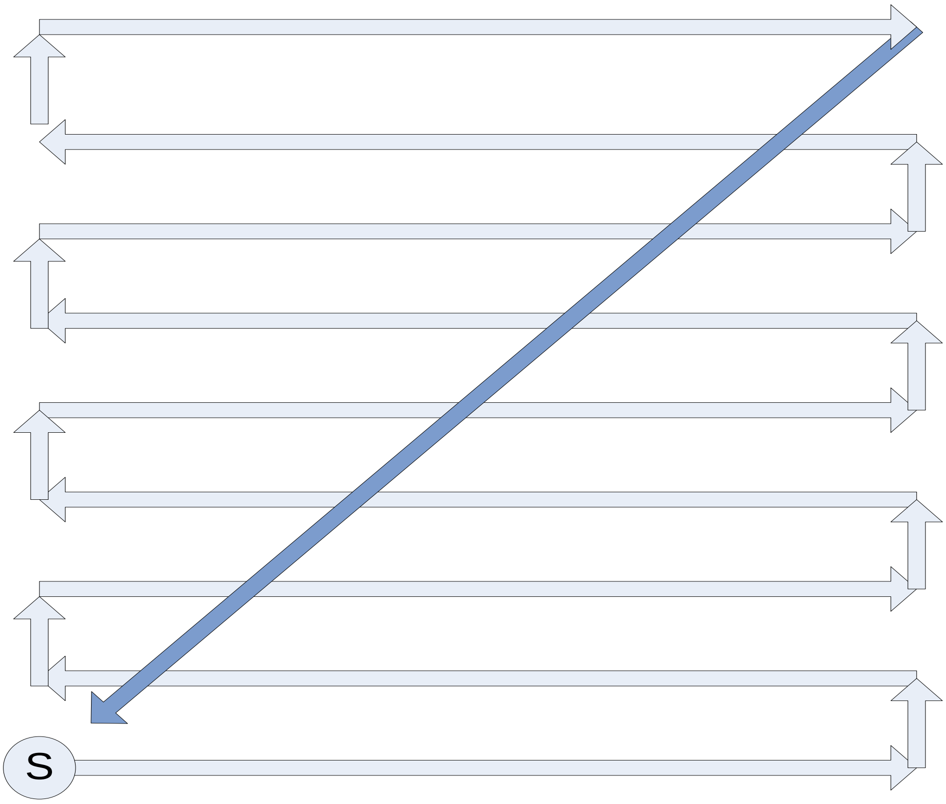


Figure 3.5 The movement pattern of BS in the simulation area

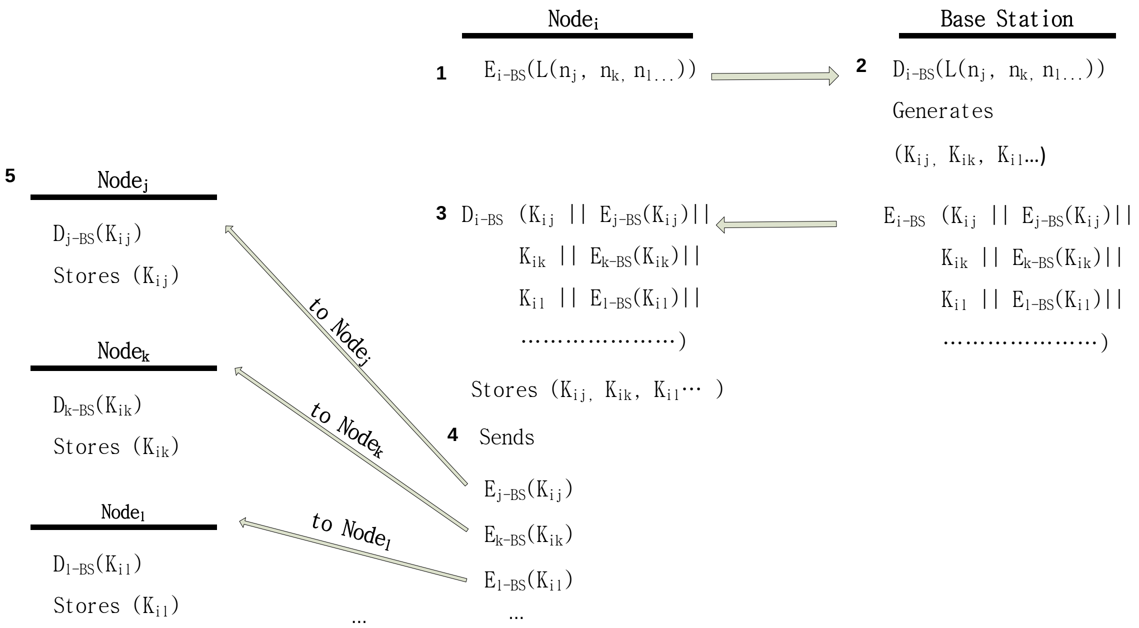
The symbols and notations we use for our scheme are listed in Table 3.1 below.

Table 3.1 List of symbols used in our scheme

n_i	A node with unique identification number *i*, node *i*
K_{i-BS}	Pairwise key shared between node *i* and BS
K_{ij}	Paiwise key shared between node *i* and node *j*
$E_{ij}(M)$	Encryption of a message *m* with pairwise key K_{ij}
$D_{ij}(M)$	Decryption of a message *m* with pairwise key K_{ij}
$L(n_j, n_k, n_l \ldots)$	List of nodes; node *j*, node *k*, node *l*…
PRF	A pseudo random function
m	The maximum number of keys a node can have.
N	The number of sensor nodes in the network
S_i	The speed of node *i*
$BS\ Speed$	The speed of Base Station

We have four important components to our scheme, namely; initialization phase, key distribution phase, shared-key discovery phase and update of the key chain. These components are explained below:

- *Initialization Phase:* This phase covers initial node configuration and node deployment to the area. Before deployment, each node i is preloaded with a unique pairwise key K_{i-BS} it shares with the Base Station. A node does not have any key it can use with other nodes, however it has a fixed key chain size m to be used later. If a node is configured to be mobile it is given an initial speed selected randomly using uniform distribution between $[speedmin, speedmax]$ and a direction between $[0, 2\pi]$. The nodes are deployed to the area using uniform distribution. After the nodes are deployed, they cannot communicate with each other until BS distributes keys to nodes. When a node meets with BS, key distribution phase begins.
- *Key Distribution Phase:* When a node senses BS in its communication range, key distribution phase starts. The flow of key distribution can be seen in Figure 3.6



$Node_i$
Base Station
1
$E_{i-BS}(L(n_j, n_k, n_l...))$
2
$D_{i-BS}(L(n_j, n_k, n_l...))$
Generates
$(K_{ij}, K_{ik}, K_{il}...)$
5
$Node_j$
$D_{j-BS}(K_{ij})$
Stores (K_{ij})
3
D_{i-BS} $(K_{ij} \| E_{j-BS}(K_{ij})\| K_{ik} \| E_{k-BS}(K_{ik})\| K_{il} \| E_{l-BS}(K_{il})\| ....)$
E_{i-BS} $(K_{ij} \| E_{j-BS}(K_{ij})\| K_{ik} \| E_{k-BS}(K_{ik})\| K_{il} \| E_{l-BS}(K_{il})\| ....)$
to $Node_j$
$Node_k$
$D_{k-BS}(K_{ik})$
Stores (K_{ik})
Stores $(K_{ij}, K_{ik}, K_{il}\cdots)$
to $Node_k$
4
Sends
$Node_l$
$D_{l-BS}(K_{il})$
Stores (K_{il})
to $Node_l$
$E_{j-BS}(K_{ij})$
$E_{k-BS}(K_{ik})$
$E_{l-BS}(K_{il})$
...
...

Figure 3.6 Key distribution protocol between base station and nodes

At *Step* 1 shown in Figure 3.6, a node *i*, who wants to get keys from BS, prepares a list $L(n_j, n_k, n_l \dots)$ of its neighbors with whom it does not share a common key. It encrypts the list with K_{i-BS}, its unique pairwise key with BS, and sends the encrypted message to BS. At *Step* 2, BS decrypts the message it got from node *i*. Using the pseudo random function PRF, BS generates pairwise keys K_{ij}, K_{ik}, K_{il} ... between n_i and the nodes in the list. For each key it created, it encrypts the key with the corresponding keys $K_{j-BS}, K_{k-BS}, K_{l-BS}$ ... it has with the nodes in the list. It concatenates the keys to be sent to n_i and these encrypted keys, creates a list of keys, encrypts it with K_{i-BS} and sends this message to n_i. At *Step* 3, n_i decrypts the message it got from BS using K_{i-BS} and gets the keys it requested from BS. It adds the keys to its key chain. For the encrypted keys sent to it, n_i sends these encrypted messages $E_{j-BS}(K_{ij}), E_{k-BS}(K_{ik}), E_{l-BS}(K_{il})$ ... to its respected receipents n_{ij}, n_k, n_l... at Step 4.At Step 5, each node decrypts the message it got from n_i using their pairwise keys they share with BS, gets the pairwise key to be used with n_i and adds the key to its key chain.

- *Shared-key Discovery Phase:* When two neighboring nodes want to communicate with each other, they first exchange their node IDs. Using these IDs they look at their key chains. If they have a pair-wise key their key chains they can communicate with each other using that key.
- *Update of the Key Chain:* As mentioned earlier, in our scheme, each node has a fixed key chain size. Therefore there needs to be an update mechanism to manage the use of this limited key chain. We use a first-in-first-out mechanism to update the key chain. In our scheme, when a node gets new keys from BS, it first checks whether it has enough space in its key chain or not. If it has enough space it adds the keys to its key chain. If it does not have enough space, then it selects the first key which is not in use at that moment and deletes it from the key chain. This way it opens up space for the new keys and adds the new keys to the key chain.

3.3 Performance Evaluation

We perform simulations to see how our proposed scheme works in various scenarios. The metrics which we look for are mainly local connectivity, which is the probability of any two neighboring nodes sharing a common key, global connectivity which is the ratio of the largest isolated component to the whole network, and resilience against node capture attacks. The details of these concepts are explained in the following sub-sections. Some parameters and system configuration for our scheme is as follows:

- The number of sensor nodes in the network is 10,000.
- The deployment area is 1,000m × 1,000m.
- Nodes are deployed with uniform distribution to the simulation area.
- The wireless communication range for each node is 40m.
- The speed of the nodes is selected randomly between 5-15 meters/minute

We performed the simulations in Visual Studio 2008 environment and used C# for coding. The results of the simulations are presented in the following subsections.

3.3.1 Local Connectivity

Local connectivity is an important metric to show the performance of the key distribution schemes. It is defined as the probability of any two neighboring nodes sharing a common key. We simulate various cases to show how our scheme performs and how the local connectivity value changes over time. The cases and results are explained below.

3.3.1.1 Local Connectivity for Different *m* Values

In order to see how the number of keys each node has affects the local connectivity, we conduct simulations for different m values and calculated the local connectivity of the network versus time. The mobility models for nodes' movement and Base Station's movement is kept as explained above. We use two network models. In the first model not all the nodes are mobile. When a node is first initialized, it has a probability of 0.5 to be mobile and move with the Random Walk Mobility Model or to stay static. We call this model Half-Mobile case. In the other model, all the nodes are mobile and that model is called Fully-Mobile case. For the m values, we use m=100, m=150, m=200 and m=250. The speed of BS is kept constant, which is 400 meters/minute. The results are shown in Figure 3.7 and 3.8.

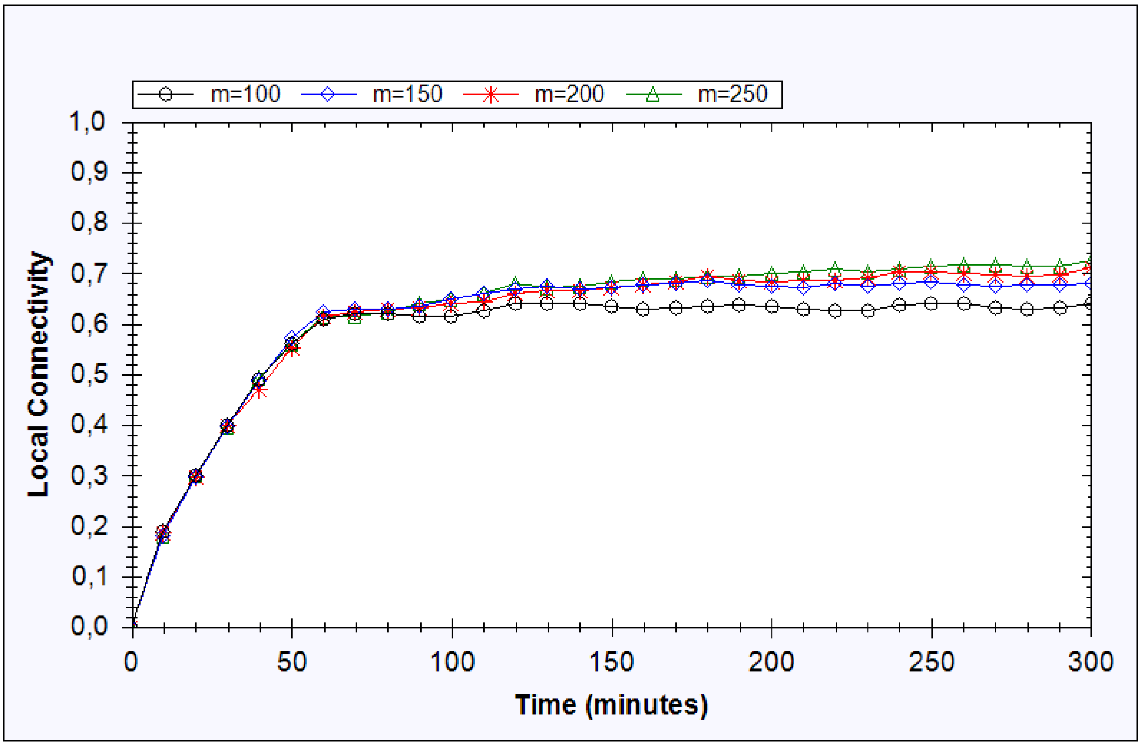


Figure 3.7 Local connectivity versus time for different *m* values for half-mobile case where *BS speed*=400 meters/minute

As it can be seen in Figure 3.7 at first local connectivity of the nodes are at 0, since no nodes are predistributed any keys apart from the key they share with Base Station. As BS starts to move and distribute keys to the nodes, local connectivity starts to increase and after a while it reaches a steady state. The time at which local connectivity reaches this value first, is around 60 minutes which is also the approximate time at which BS completes one round of its movement in the simulation area. By this time it has moved in the simulation area and covered the area completely, distributing pairwise keys to the nodes it encountered along the way. If all the nodes were static, local connectivity value would have been 1, since all nodes would get the pairwise keys for their neighbors and neighboring relations would not ever change after that. However, as mentioned earlier, half of the nodes are mobile; therefore neighboring relations keeps changing, thus the connectivity does not reach a very high value.

Local connectivity for *m*=100 is around 0.64, for *m*=150 it is 0.68, for *m*=200 it is 0.71 and for *m*=250 local connectivity is around 0.73. These results show that the change in the value of *m* does not bring a big difference in terms of local connectivity. The reason for that is the following: Base Station as a key distribution center provides nodes with keys of their immediate neighbors at that time and nodes update their key chains with their newly acquired keys by the method described earlier. This means, they do not keep keys with their old neighbors. Also the keys which are really useful for the nodes are their freshest keys with their immediate neighbors, since local connectivity is about the connectivity between neighboring keys. Therefore, the other keys they keep in their memory has little help to them in terms of local connectivity since they become useless as the neighboring relations keep changing and old neighbors move away from each other. So the decisive element in terms of local connectivity for this case is not the size of the key chains, but the number of neighbors a node can have around it. In our simulations the maximum and average numbers of neighbors for a node is approximately 90 and 60, respectively.

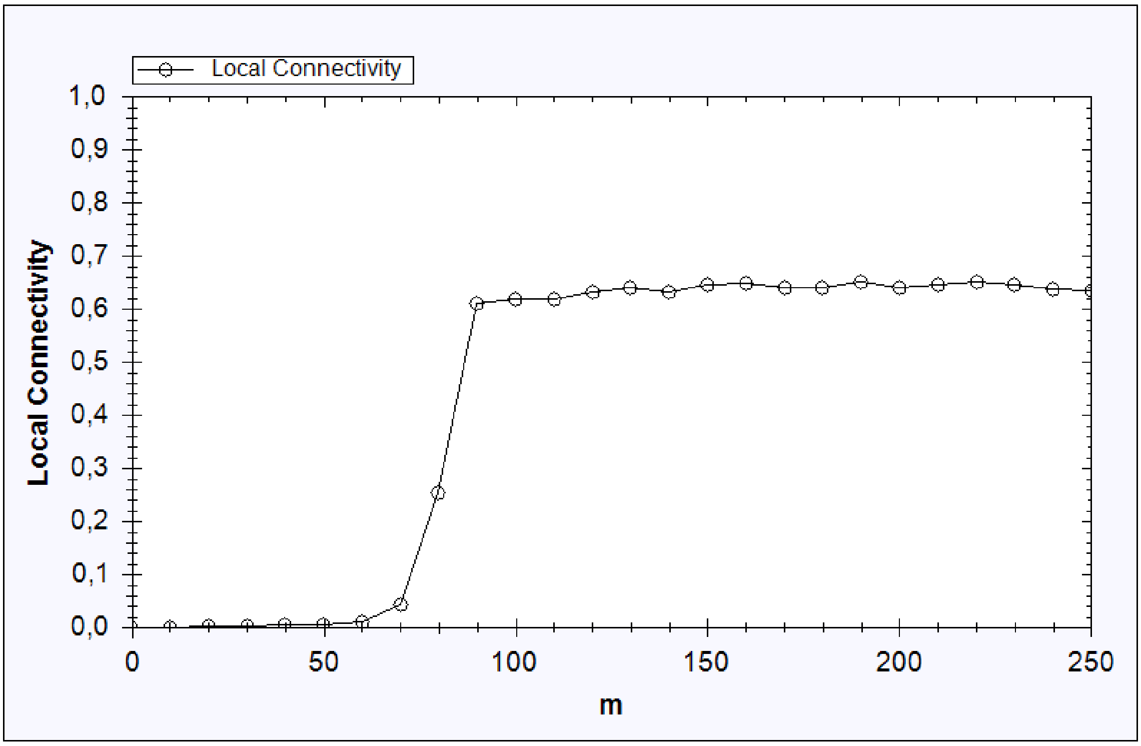


Figure 3.8 Local connectivity versus *m* values for half-mobile case where *time*=120 minutes

Figure 3.8 shows the local connectivity values versus *m* for half-mobile case. In this simulation, nodes first move in the environment for 120 minutes, and then their local connectivity is calculated. Figure 3.8 shows that local connectivity value reaches a convergence value around *m*=90 which shows that the discussion above is valid.

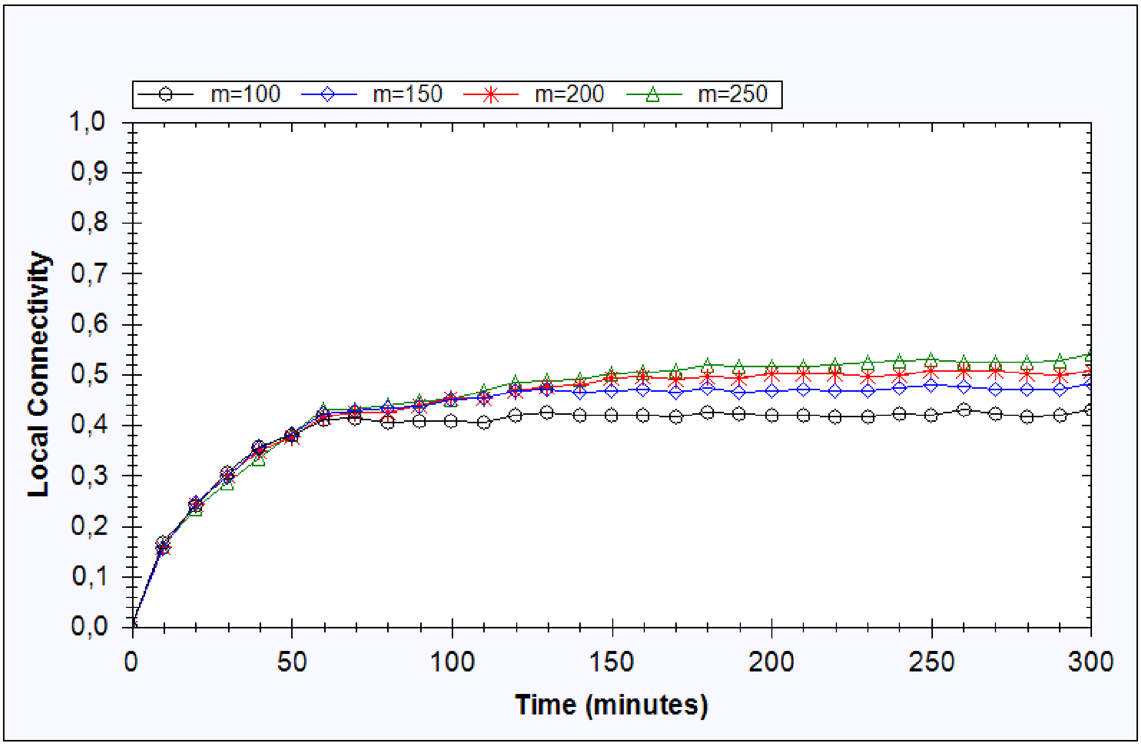


Figure 3.9 Local connectivity versus time for different *m* values for fully-mobile case where *BS speed*=400 meters/minute

Figure 3.9 shows the local connectivity for a fully mobile network. The immediate difference from half-mobile case is the decrease in local connectivity values. This is an expected result because all the nodes are mobile in this case; therefore, neighboring relations change faster than half-mobile case. Thus the effect of BS distributing keys to the neighbors does not last long and connectivity stays lower than half-mobile case. Local connectivity values for *m*=100 is around 0.44, for *m*=150 it is 0.48, for *m*=200 it is 0.51 and for *m*=250 local connectivity is around 0.54. The convergence time is again around 60 minutes and the difference of local connectivity values for different *m* values is almost the same as half-mobile case for which the reasons are already explained above.

3.3.1.2 Local Connectivity for Different BS Speeds

In this section we discuss how the speed of BS affects the local connectivity of the network. In order to observe this, we run simulations using different BS speeds for half-mobile and fully-mobile case. In these scenarios, the size of the key chain *m* is kept constant at *m*=200. BS speed, on the other hand, has three different values; *BS speed*=200 meters/minute, *BS speed*=400 meters/minute and *BS speed*=600 meters/minute. Figure 3.10 and Figure 3.11 show local connectivity versus time, for half-mobile and fully-mobile cases.

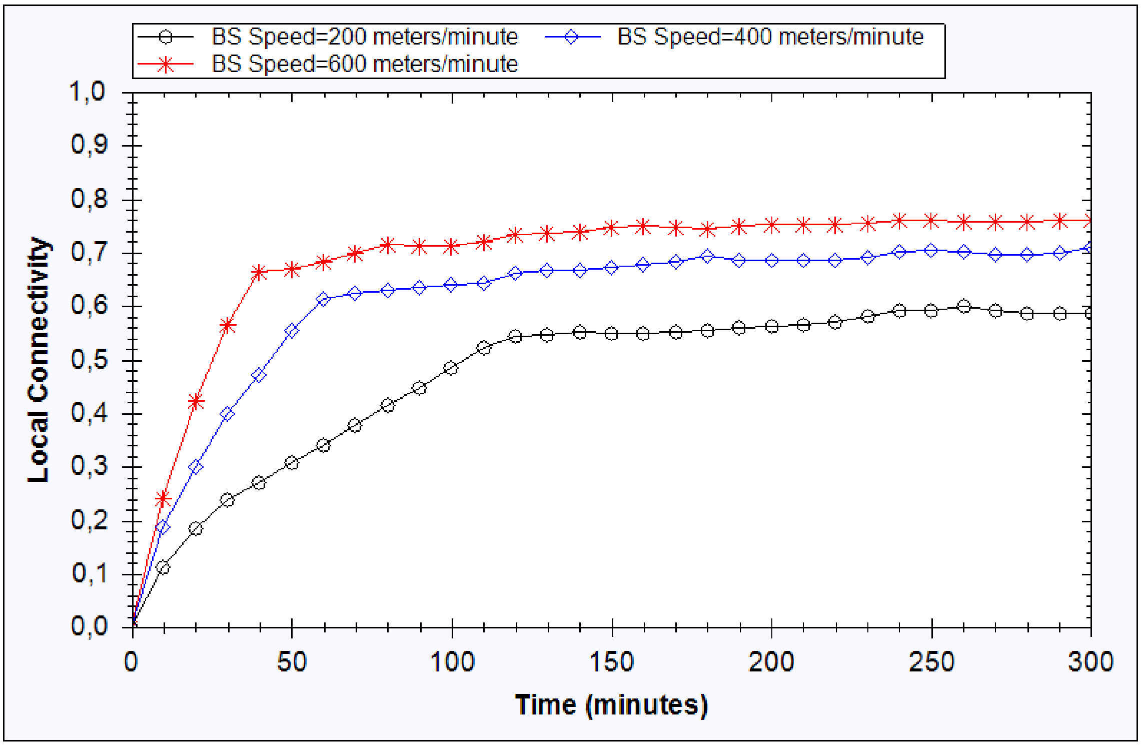


Figure 3.10 Local connectivity versus time for different BS speeds for half-mobile case where *m*=200

Figure 3.10 shows that the speed of the BS affects local connectivity in various ways. Firstly, the convergence time for local connectivity decreases as BS speed increases. For *BS speed*=600, local connectivity reaches its convergence value around *time*=40 minutes, for *BS speed*=400, it reaches the convergence value at *time*=60 minutes and for *BS speed*=200, it reaches the convergence value around *time*=120 minutes. The reason for this difference is that when BS moves faster, it can cover the whole simulation area faster and connect the neighboring nodes with each other. As it was explained in the previous sections the time at which local connectivity first reaches its convergence value is the time BS completes its one round of scanning in the area. With faster speeds, the round is completed faster; thus the convergence occurs earlier compared to the slower speeds. Secondly, local connectivity value increases when BS moves faster. For *BS speed*=600, local connectivity is around 0.78, for *BS speed*=400 it is around 0.71 and for *BS speed*=200 local connectivity is around 0.59. The reason for this change is as follows. When BS moves faster, the expected time it meets a node again and updates its key chain is shorter as compared to BS moving slower. Therefore it can update the nodes with their neighboring nodes' keys more frequently when it is faster and keep the connectivity higher than the slower cases.

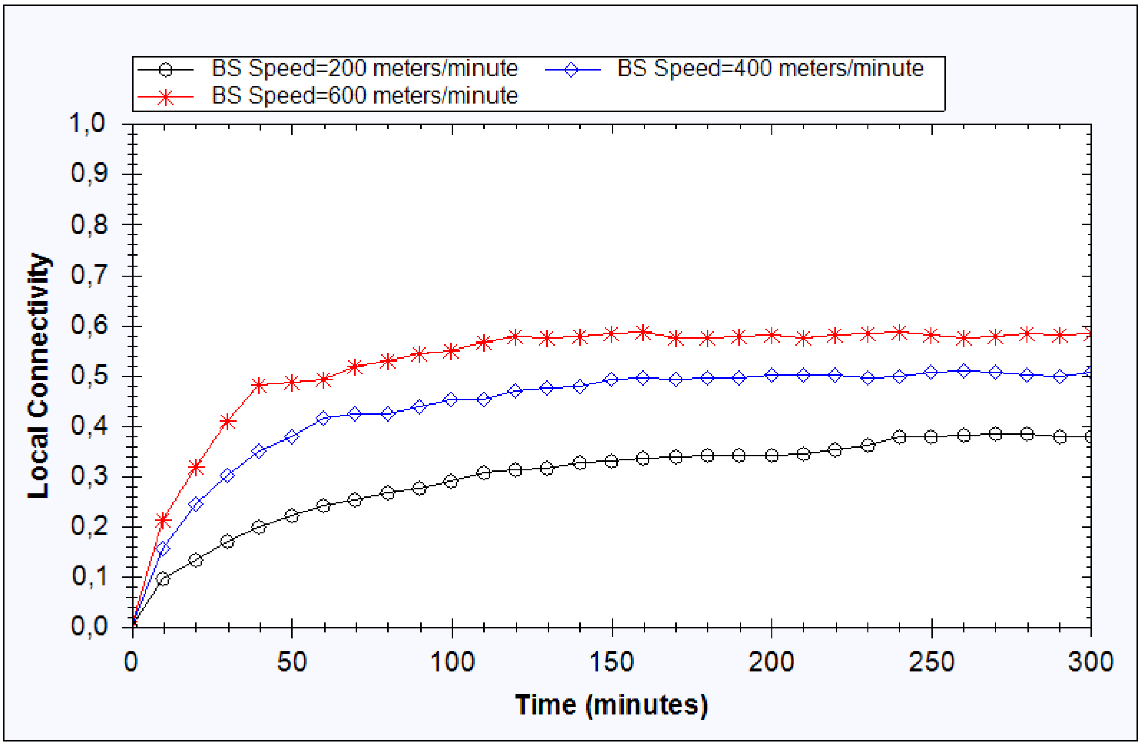


Figure 3.11 Local connectivity versus time for different BS speeds for fully-mobile case where m=200

For fully-mobile case, it can be seen that local connectivity values are lower than half-mobile-case. The reason is for that is the faster change in neighboring relation between nodes, since they move away from each other faster which makes more of their keys in the key chains useless after a while. However, just like the half-mobile case, local connectivity increases as BS speed increases. Also it can be seen that the time the networks reaches its convergence value in terms of local connectivity is the same as the half-mobile case.

3.3.1.3 Local Connectivity Using Multiple Base Stations

In the above subsections for both cases only one Base Station is used. In this section we discuss how it affects local connectivity when multiple Base Stations as key distribution centers are used. In order to see the difference, we use a scenario where there are two BSs in the simulation area moving at the same time. In this case similar to the single BS movement explained before, the BSs move in the area deterministically and cover the whole area, however one BS moves in the lower half of the simulation area and distributes keys to nodes in that area; while the other BS moves in the upper half of the area and distributes keys in that area. Figure 3.12 shows the movement of two BSs in the simulation area.

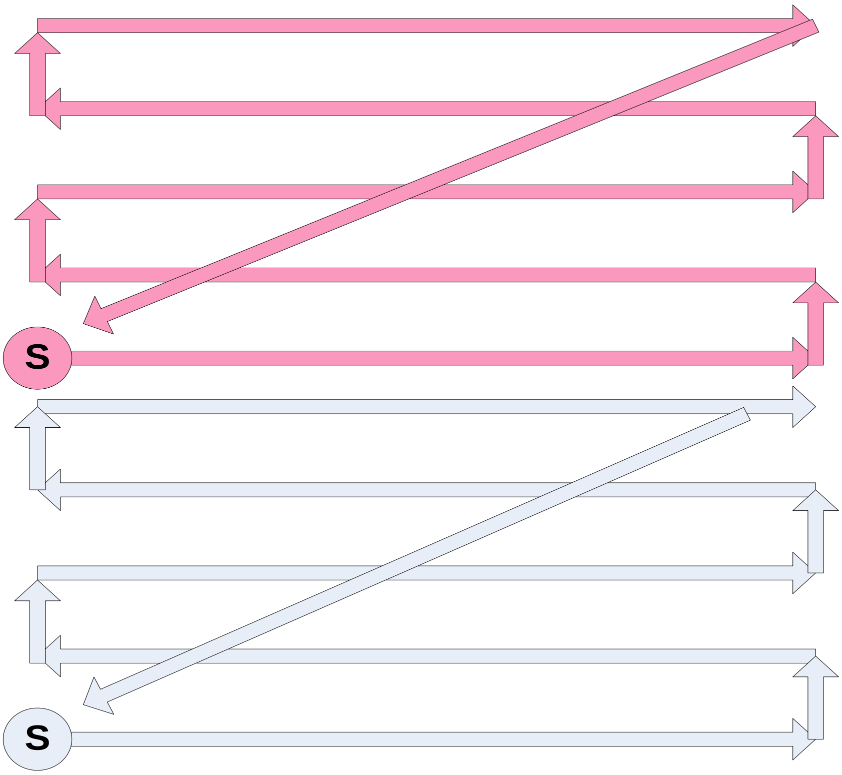


Figure 3.12 The movement pattern of two base stations in the simulation area

We conduct simulations using two BSs and compare its results to the case when we use only one BS. The speed of BS is kept constant at 400 meters/minute. The size of *m* is also kept constant at *m*=200. The simulation is run for both half-mobile and fully-mobile cases.

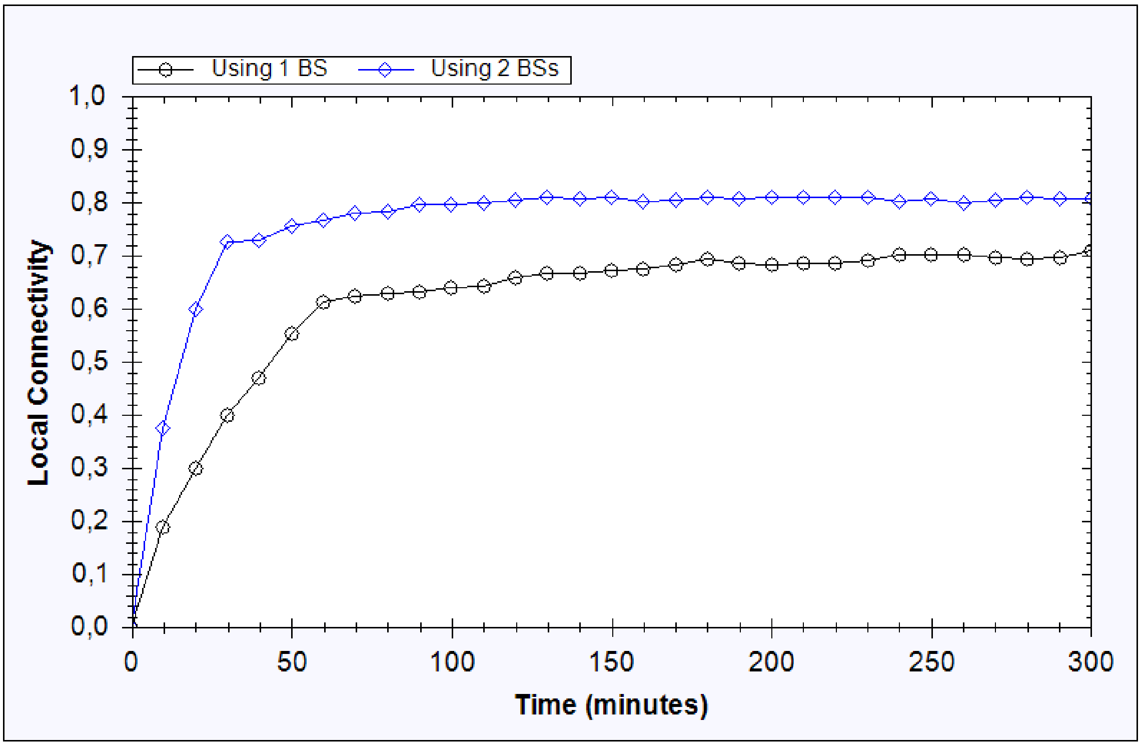


Figure 3.13 Local connectivity versus time using multiple BSs for half-mobile case

Figure 3.13 shows the local connectivity values using one BS versus using two BSs for half-mobile case. It is seen in the figure that local connectivity where two BSs is used is higher than the scenario where only one BS is used. The reason is similar to the previous case where the speed of BS increased. In this case as well, a node meets a BS more frequently if there are two BS in the simulation area and gets its key chain updated more frequently, therefore local connectivity reaches a higher value. Also the time needed for the local connectivity to reach its convergence value is shorter in two BSs case. When there are two BSs in the environment, the whole area gets covered in 30 minutes and local connectivity reaches its convergence value at that time. When there is only one BS in the environment, the area gets covered in 60 minutes; hence convergence in local connectivity is reached around that time as well.

Similar results can be seen in fully-mobile network except that the local connectivity value is lower than the half-mobile case. Note that the convergence time for both two BSs case and one BS case stays the same, since this time is related to BS speed which is the same for half-mobile and fully-mobile cases but not to the mobility of the nodes.

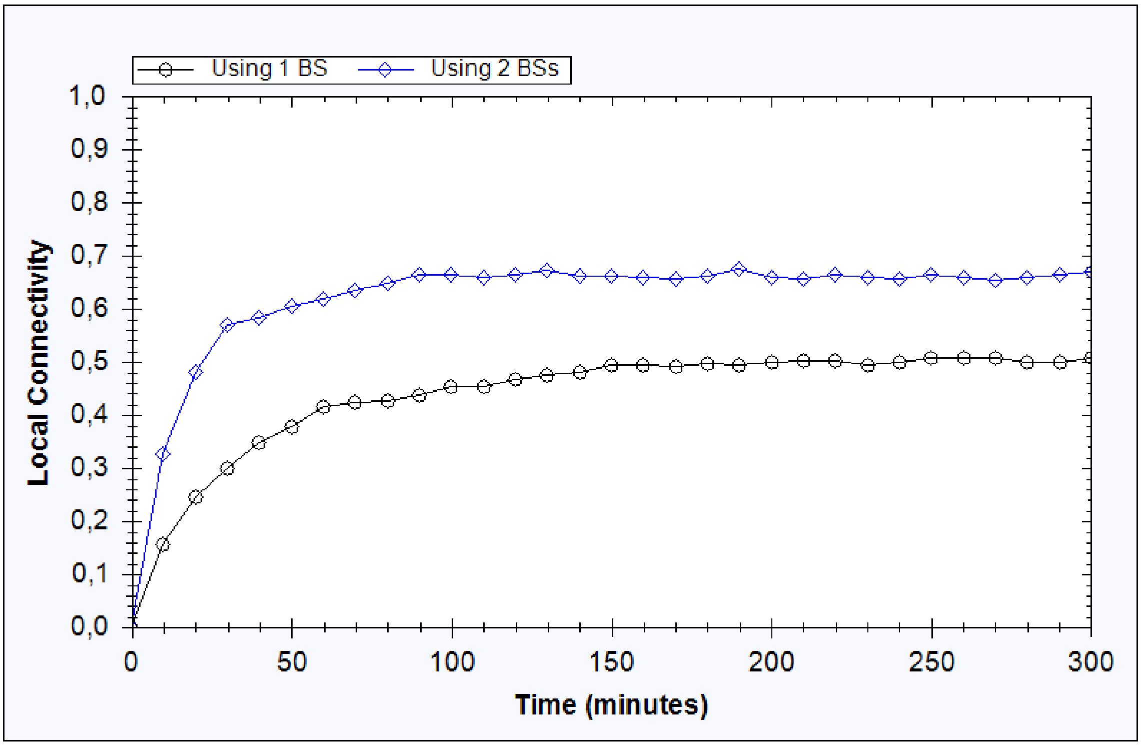


Figure 3.14 Local connectivity versus time using multiple BSs for fully-mobile case.

3.3.1.4 Local Connectivity When BS Stops After One Round

In our original scheme, BS moves continuously and distributes keys throughout the life of the network. In this case however, BS stops its movement and ceases key distribution once it finishes its one round in the simulation area. In order to see the performance of this case, we conduct simulations for both half-mobile and fully-mobile cases. We keep *m* constant at 200, *BS speed* constant at 400 meters/minute (during its one round of movement only). The results are shown in Figure 3.15.

Figure 3.15 shows that after BS stops its movement and key distribution, local connectivity value starts to decrease gradually. Local connectivity decreases more for fully-mobile case compared to half-mobile case. This is an expected behavior since in fully-mobile case neighboring relationships change faster than half-mobile case and keys become useless faster.

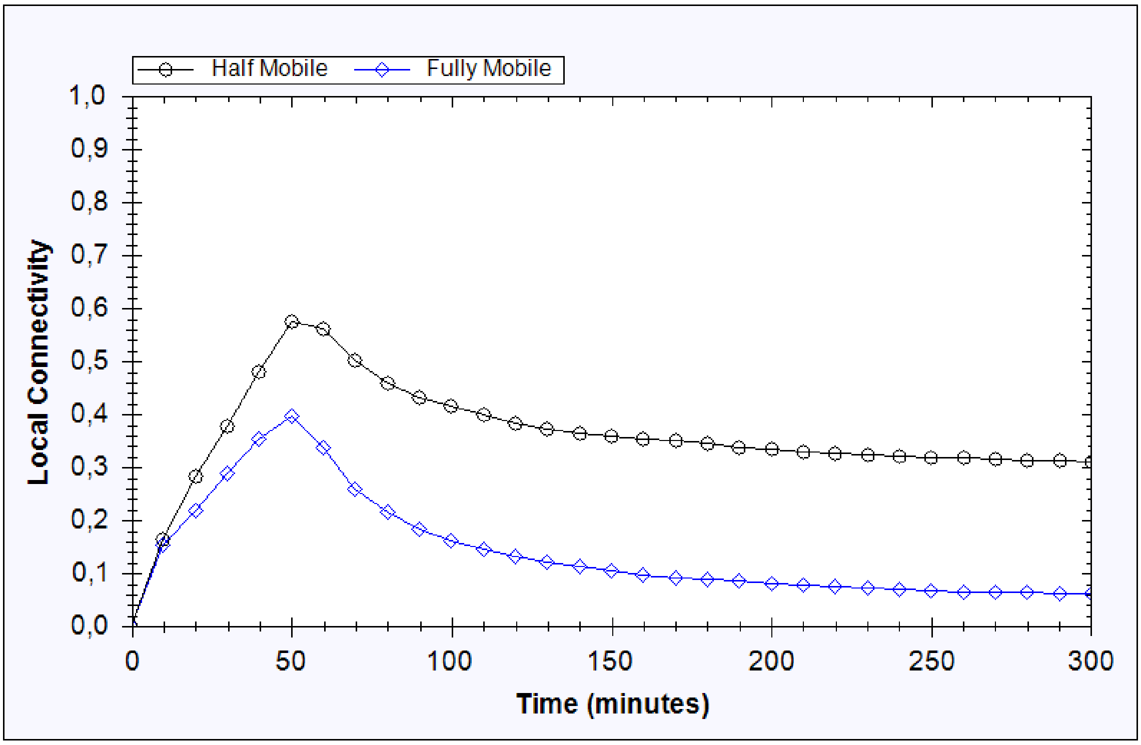


Figure 3.15 Local connectivity versus time for half-mobile and fully-mobile cases when BS stops movement and key distribution after completing one round

3.3.1.5 Local Connectivity for Different Communication Ranges

In the simulation for the previous subsections communication range of nodes is kept constant at 40 meters. In this case, we use different communication ranges which are 20 meters, 40 meters and 60 meters and observe how local connectivity changes. In our simulations we use *m*=200, *BS speed*=400 meters/minute. The results of the simulation for half-mobile and fully-mobile cases are shown in Figures 3.16 and 3.17 respectively.

Figure 3.16 shows that local connectivity increases for higher communication ranges because as communication range increases, the area in which the nodes are connected by BS increases. Figure 3.17 shows similar relative behavior, only local connectivity value is lower due to limited mobility.

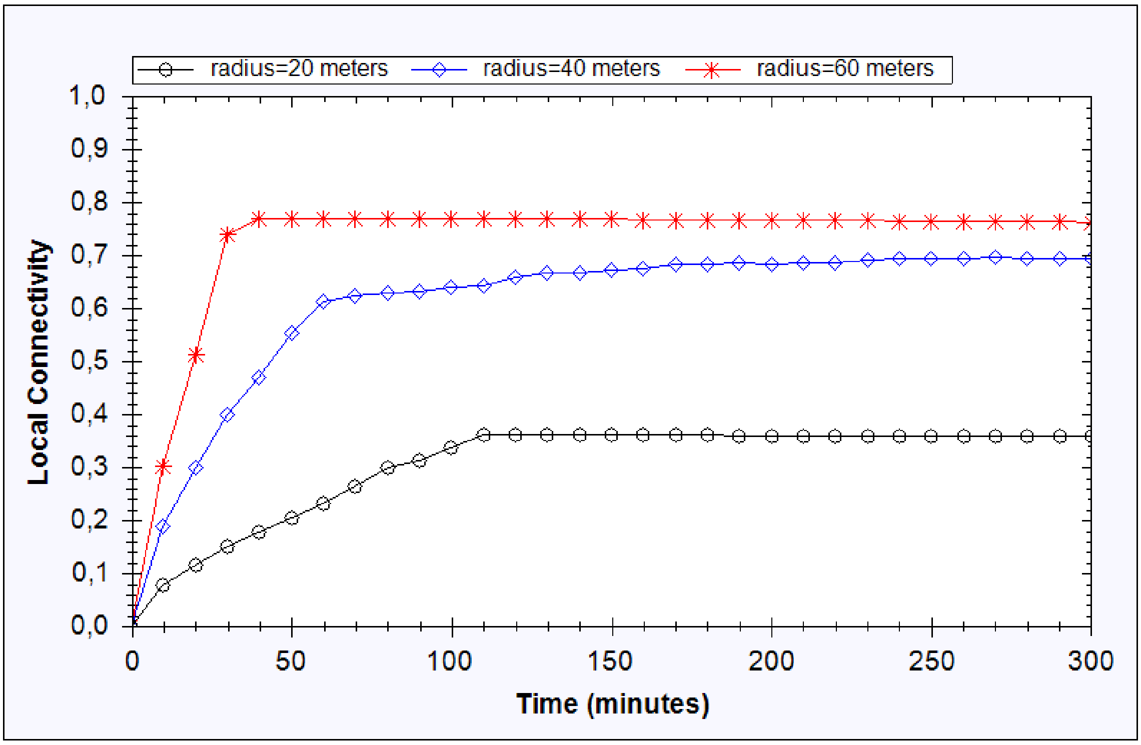


Figure 3.16 Local connectivity versus time for different communication ranges for half-mobile case where *m*=200

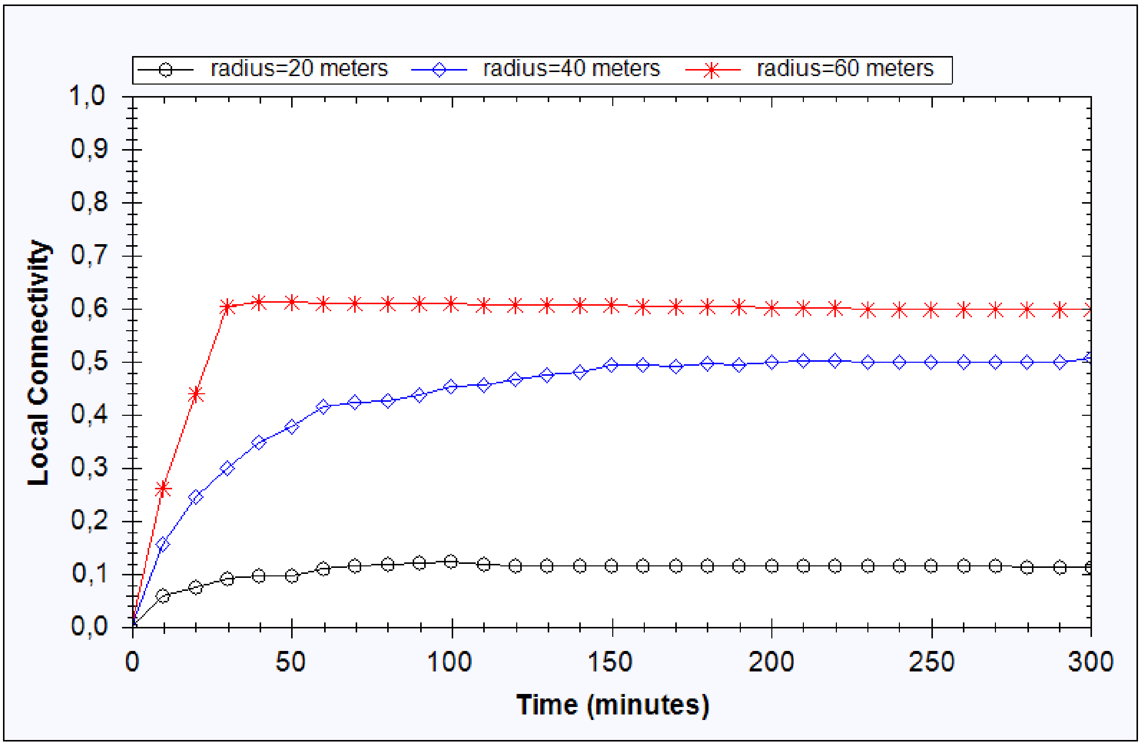


Figure 3.17 Local connectivity versus time for different communication ranges for fully-mobile case where *m*=200

3.3.1.6 Local Connectivity using Multiple Static Base Stations

Our original scheme uses a mobile BS which acts as a key distribution center. In this case, we use static BSs in the environment and conduct simulations to see local connectivity performance. BSs are deployed to the environment such that they do not overlap with each other. Since they are static, only the nodes that come into the communication range of a BS can get keys from them. In our simulation we use 25 BSs, 49 BSs and 100 BSs in the environment. With 25 BSs we cover 12.5%, of the sensor field. Similarly with 49 BSs we cover 25% and with 100 BSs we cover 50% of the sensor field. We keep *m* constant at 200. The results for half mobile and fully-mobile cases are shown in Figure 3.18 and Figure 3.19.

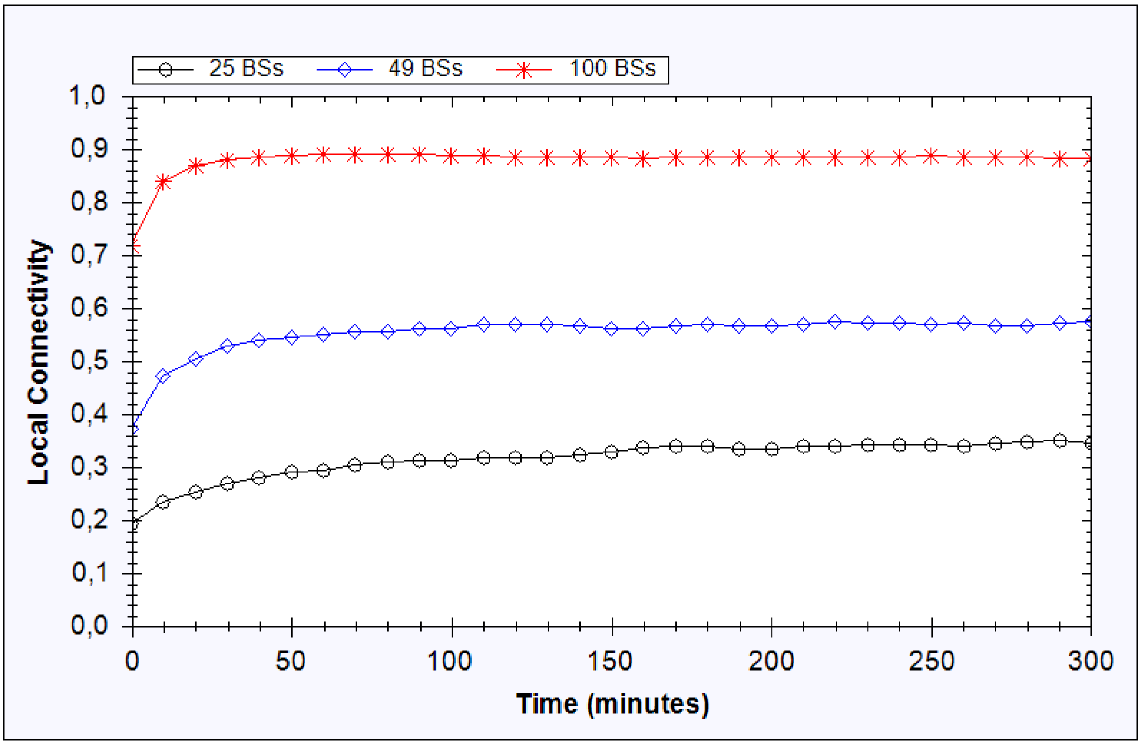


Figure 3.18 Local connectivity versus time using multiple static BSs for half-mobile case where *m*=200

Figure 3.18 shows that local connectivity increases when we use more BSs in the environment. Local connectivity starts from 0.19 for 25 BSs, 0.37 for 49 BSs and 0.72 for 100 BSs. After simulation starts, local connectivity increases because nodes move and come into vicinity of BSs and get their key chains updated. Figure 3.19 shows similar relative behavior, but local connectivity values are higher than half-mobile case for all three numbers of BSs. The reason is that all the modes are mobile in this case, which results in more nodes coming into contact with BSs and getting keys from them.

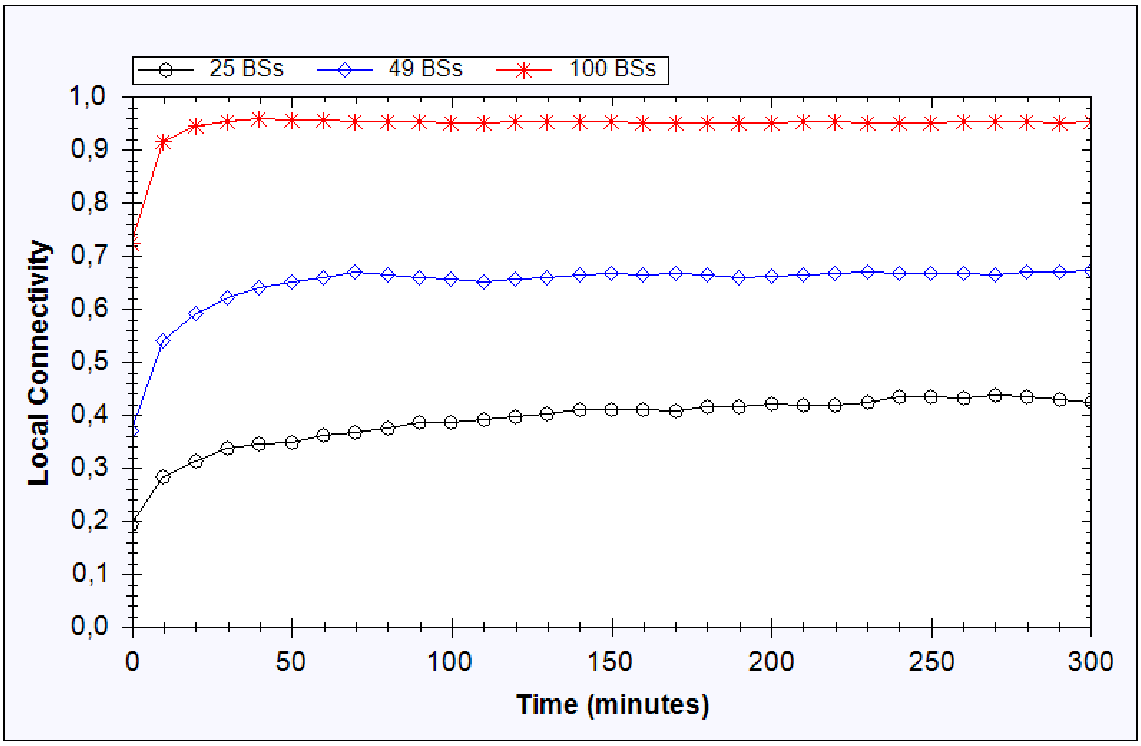


Figure 3.19 Local connectivity versus time using multiple static BSs for fully-mobile case where m=200

3.3.2 Global Connectivity

Global connectivity is another important performance for key distribution schemes. Let G be a key sharing graph with nodes as its vertices and the secure links between nodes (i.e. the links between nodes which share a key) as its edges. Global connectivity is the ratio of the size of the largest component in G to the size of the whole network [3]. It is important for a network to have a global connectivity, because if the global connectivity value is high it means the network connectivity is high, and nodes can communicate with each other even if they have to use a few hops. We conduct simulations for various cases to see how our scheme performs in terms of global connectivity. The next subsections explain each case and the corresponding results.

3.3.2.1 Global Connectivity for Different *m* Values

Similar to the simulations we conducted for local connectivity, we first conduct simulations to see how the change in key chain size affects the global connectivity. For this purpose we use different values for *m*, namely 100, 150, 200 and 250 and calculate global connectivity versus time. The nodes move with Random Walk Mobility Model and Base Station moves by the deterministic mobility model explained before. BS speed is kept constant at 400 meters/minutes. We run simulations for both half-mobile and fully-mobile cases. The results are shown in Figure 3.20 and 3.21.

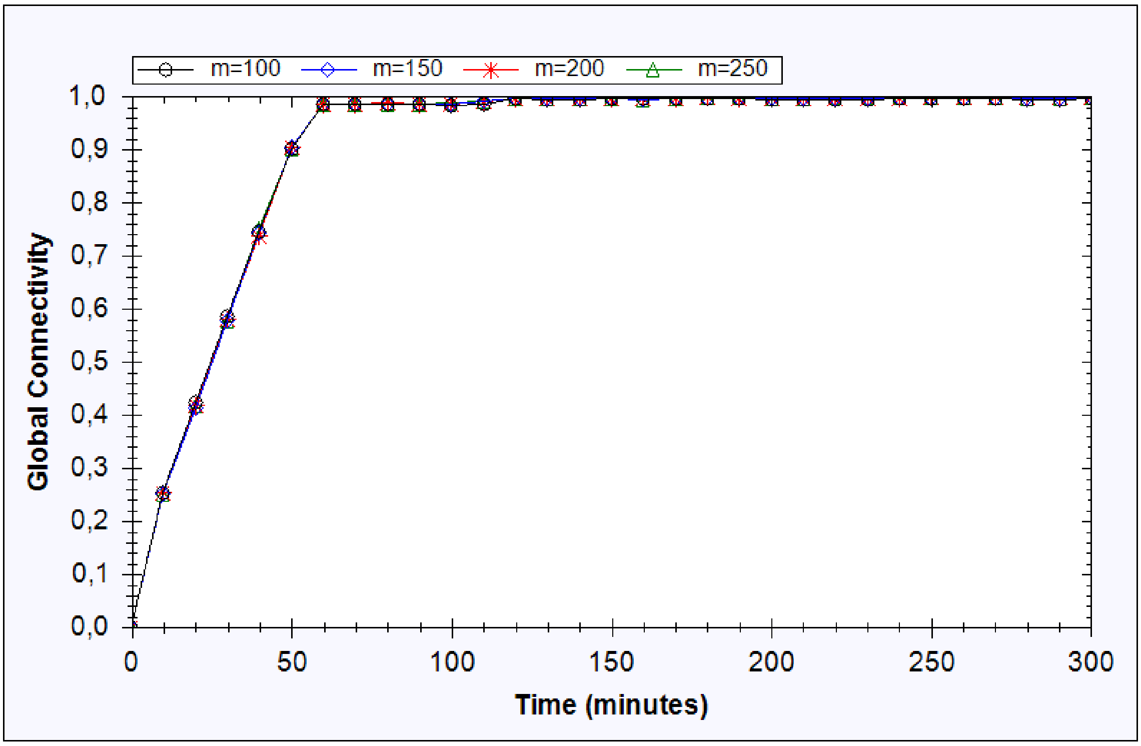


Figure 3.20 Global Connectivity versus time for different *m* values for half-mobile case for *BS speed*=400 meters/minute

It can be seen in Figure 3.20 that global connectivity reaches its convergence value, which is close to 1, at *time*=60. This is the time BS completes its one round of movement in the simulation area. After this time, the connectivity value slowly increases and eventually reaches values very close to 1. It can also be seen that there is not a significant difference between different *m* values in terms of global connectivity.

This shows that even a key chain of size 100 is enough to achieve a global connectivity which is fairly close to 1.

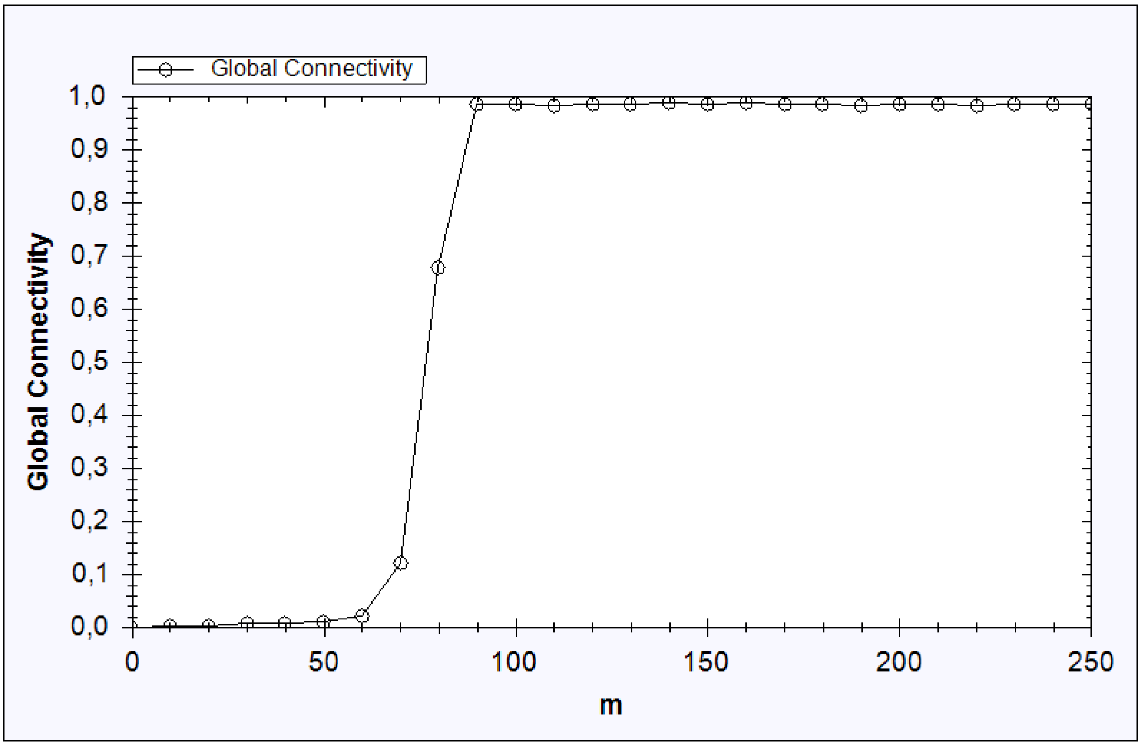


Figure 3.21 Global connectivity versus *m* values for half-mobile case where *time*=120 minutes

Figure 3.21 shows global connectivity versus *m* values after nodes move in the environment for 120 minutes. This figure also shows that the decisive element in global connectivity is the maximum number of neighbors of a node, which is around 90.

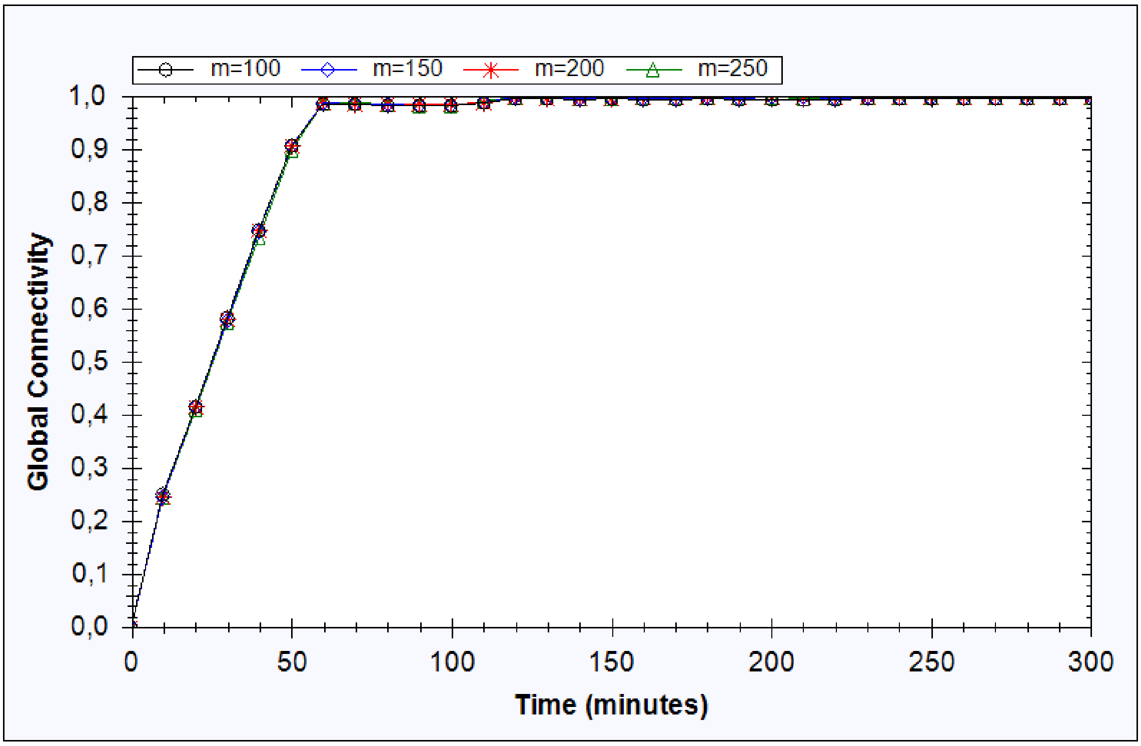


Figure 3.22 Global Connectivity versus time for different *m* values for fully-mobile case for *BS speed*=400 meters/minute

Figure 3.22 shows the global connectivity values for different *m* values for fully-mobile case. This figure is almost the same as the half-mobile case. The pattern of the increase in global connectivity and the values it reaches are almost the same as Figure 3.20. Yet, there is a little difference between values of each case. Table 3.2 shows the global connectivity values for each *m* value at *time*=300 to show that difference between fully-mobile case and half-mobile case.

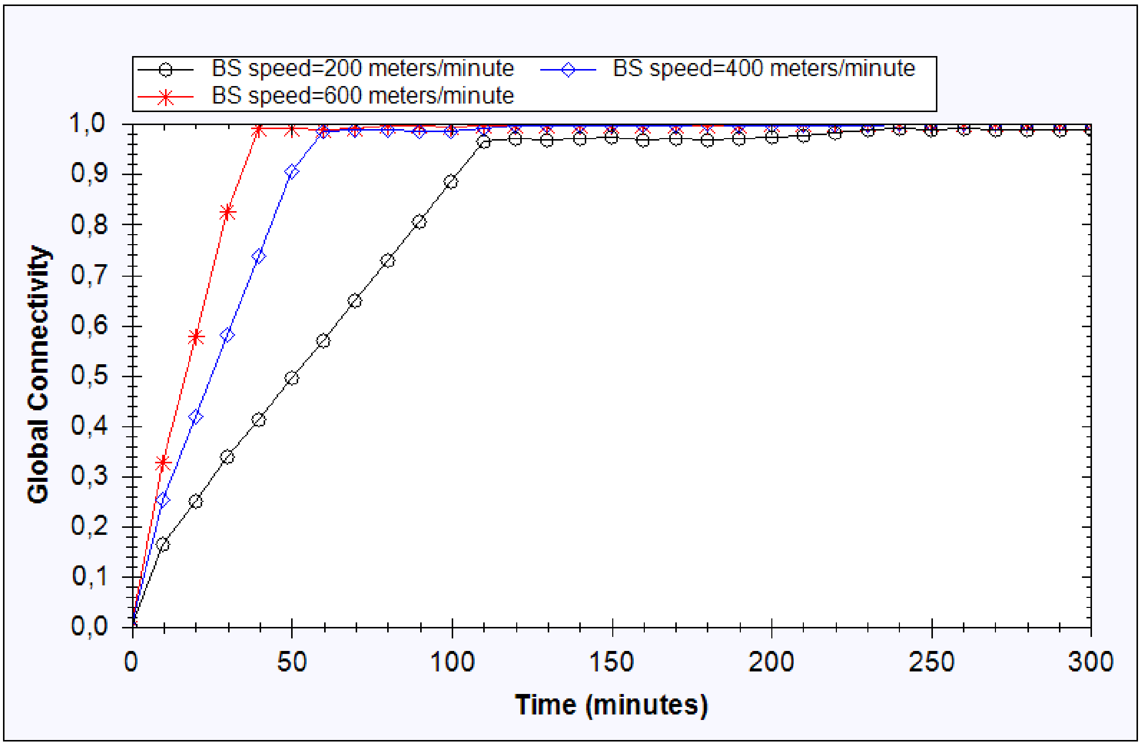
Table 3.2 Global connectivity values for half-mobile and fully mobile-cases for different *m* values at *time*=300

Key Chain Size, *m*	**Half-Mobile**	**Fully-Mobile**
m=100	0.9955	0.9960
m=150	0.9980	0.9984
m=200	0.9983	0.9986
m=250	0.9985	0.9993

Although it cannot be distinguished in the figures clearly, Table 3.2 shows that there is a subtle increase in global connectivity as *m* gets higher. However this increase is very little and even *m*=100 provides a very good global connectivity. There is also a difference between half-mobile and fully-mobile cases. It can be seen that fully-mobile case achieves a higher global connectivity, which is again a very little increase. The reason for this difference is that in half-mobile case some of the static nodes can geographically get isolated from other nodes as their neighbors move away from them and this results in this case having a lower value for global connectivity.

3.3.2.2 Global Connectivity for Different BS Speeds

Having seen the global connectivity results for different *m* values, we next conduct simulations for different BS speeds. The BS speeds we use are 200 meters/minute, 400 meters/minute and 600 meters/minute. The value of *m* on the other hand is kept constant at 200. Figure 3.23 shows the global connectivity versus time for fully-mobile case.



3.23 Global connectivity versus time for different BS speeds for fully-mobile case where *m*=200

Figure 3.23 shows that as BS speed increases, the time needed for global connectivity value to reach its convergence value decreases. For *BS speed*=600, this value is reached at *time*=40, for *BS speed*=400, the convergence value is reached at *time*=60 and for *BS speed*=200 it is reached at around *time*=110. It can also be seen that for all three cases, after simulation area is covered by BS wholly at the mentioned times, global connectivity value slowly increases and eventually becomes very close to 1. The same simulation was run for half-mobile case as well. Since the pattern of that case is the same as fully-mobile case, we did not put that figure here, however Table 3.3 shows the global connectivity value for different BS speeds at *time*=100, *time*=200, and *time*=300 for both half-mobile case and fully-mobile case.

Table 3.3 Global connectivity values for half-mobile and fully mobile case for different BS speeds at *time*=300

BS speed	**Time**	**Half-mobile**	**Fully Mobile**
BS speed=200 meters/minutes	100	0.8848	0.8934
	200	0.9750	0.9710
	300	0.9878	0.9930
BS speed=400 meters/minutes	100	0.9857	0.9845
	200	0.9965	0.9980
	300	0.9983	0.9986
BS speed=600 meters/minutes	100	0.9942	0.9957
	200	0.9986	0.9997
	300	0.9995	0.9997

Table 3.3 shows that, as BS speeds increases global connectivity value also increases, however this difference between different BS speeds is very little. As Figure 3.16 shows, the main difference between these three cases are the time needed to achieve global connectivity reach a value close to 1. Table 3.3 also shows that fully-mobile case has a higher global connectivity value in general compared to half-mobile case. Again this difference is also very little for which the reasons were explained in the previous subsection.

3.3.2.3 Global Connectivity for Using Multiple Base Stations

In this case we conduct simulations to see global connectivity using two BSs compare the results of using only one BS. The movement of the BSs in two BS case is the same as the case in section 3.2.2.1 and it is shown in Figure 3.12. The global connectivity value for fully-mobile case is shown in Figure 3.24. The speed of BS for both cases is kept constant at 400 meters/minutes and the key chain size of the nodes is kept constant at 200.

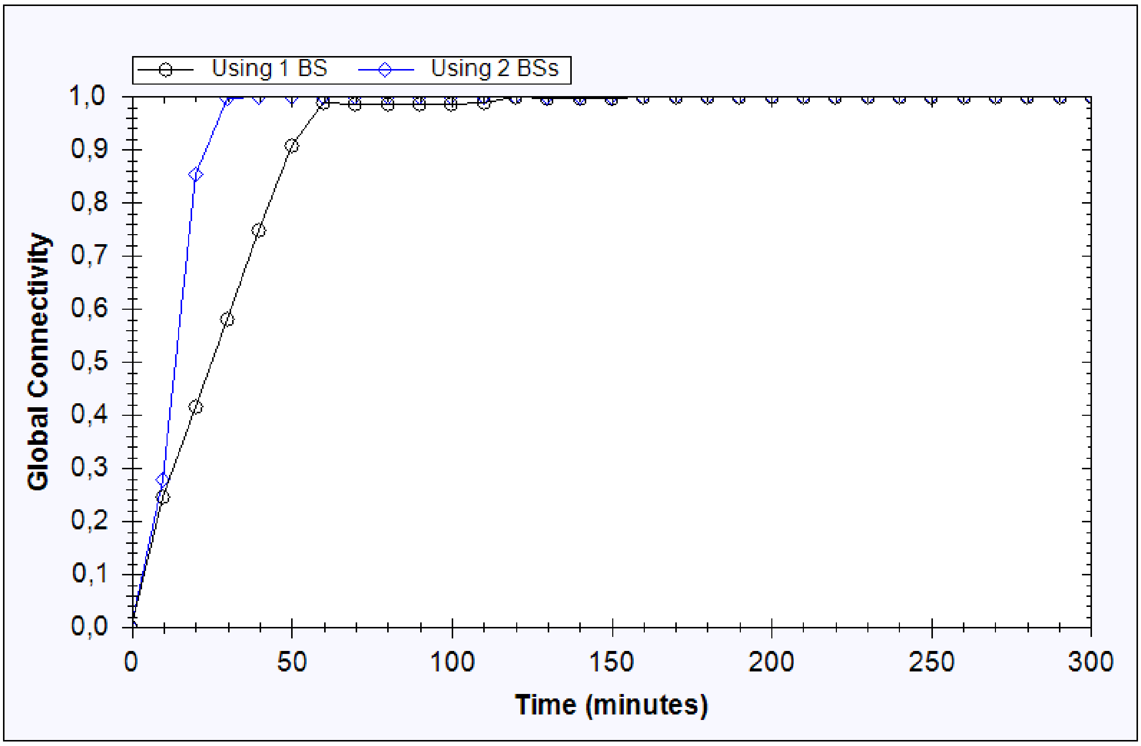


Figure 3.24 Global connectivity versus time using multiple BSs for fully-mobile case for *m*=200

It can be seen in Figure 3.24 that when there are two BS moving in the simulation area, global connectivity reaches a value very close to one first at *time*=30. When there is only one BS moving in the area, global connectivity reaches a value near 1 at around *time*=60. The same simulation is conducted for half-mobile case and the same pattern is seen in that case as well. Table 3.4 shows the global connectivity values for one BS case and two BSs case for both half-mobile case and fully-mobile case.

Table 3.4 Global connectivity values for half-mobile and fully mobile case using multiple BSs at different times

BS number	Time	Half-mobile	Fully Mobile
Using 1 BS	100	0.9857	0. 9845
	200	0.9965	0.9980
	300	0.9983	0.9986
Using 2 BS	100	0.9997	0.9997
	200	0.9996	1
	300	0.9995	0.9999

As Table 3.4 shows, global connectivity value using two BSs is higher than global connectivity value using one BS. Also, it can be seen that global connectivity value for fully-mobile case is again slightly higher than half-mobile case.

3.3.2.4 Global Connectivity When BS Stops After One Round

In this case, we stop BS after it completes its one round of movement in the environment. After that time, BS does not move and does not distribute keys to nodes. We conduct simulations to see how global connectivity changes in this case. Figure 3.25 show the global connectivity for both half-mobile and fully-mobile cases where *m* is kept constant at 200. It can be seen in the figure that global connectivity starts to decrease after BS stops its operation. As the simulation continues it is seen that fully-mobile case's global connectivity gets lower than half-mobile case's value. This is because neighboring relationships change faster in fully-mobile case since all the nodes are mobile.

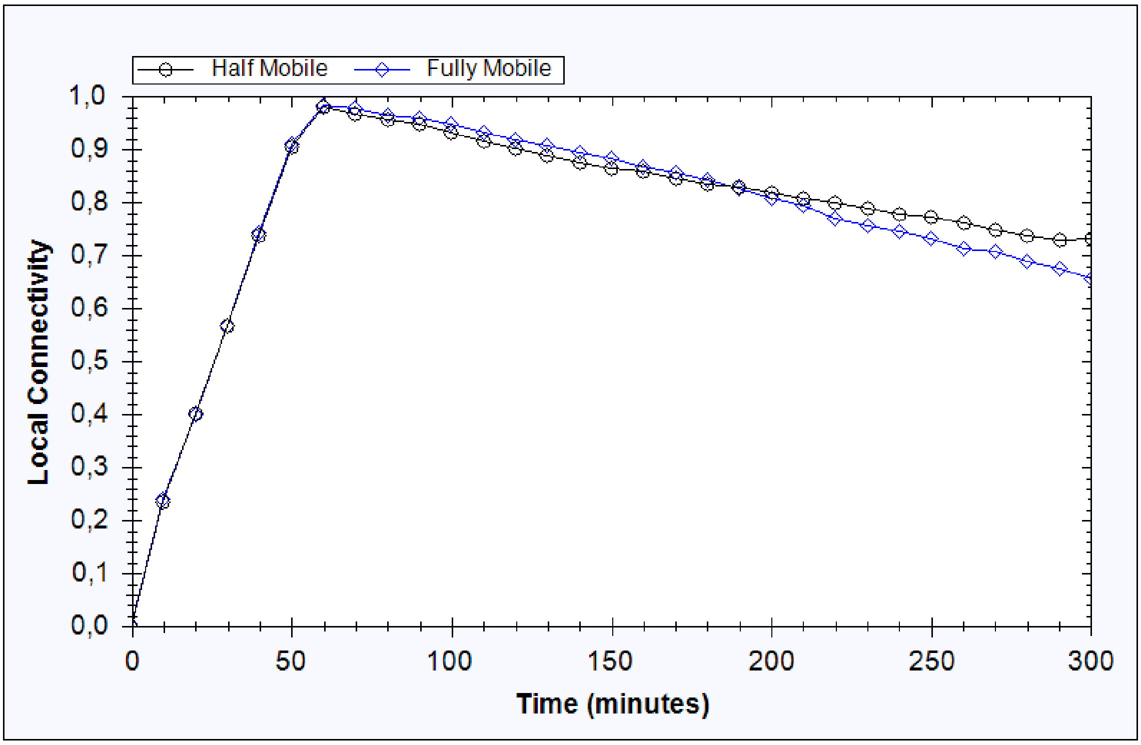


Figure 3.25 Global connectivity versus time for half-mobile and fully-mobile cases when BS stops movement and key distribution after completing one round

3.3.2.5 Global Connectivity for Different Communication Ranges

In order to see how communication range affects our scheme's performance, we conduct simulations using different communication ranges and calculate global connectivity. Simulations are conducted for both half-mobile and fully-mobile cases where communication range is 20 meters, 40 meters and 60 meters. We keep *m* constant at 200 and BS speed constant at 400 meters/minute. The results are shown in Figure 3.26 and Figure 3.27. It can be seen in the figures that global connectivity increases as communication range increases, since more nodes can get keys from BS if their communication range is higher. Also it can be seen that half-mobile case's values are higher in this case as well, for which the reasons were explained before.

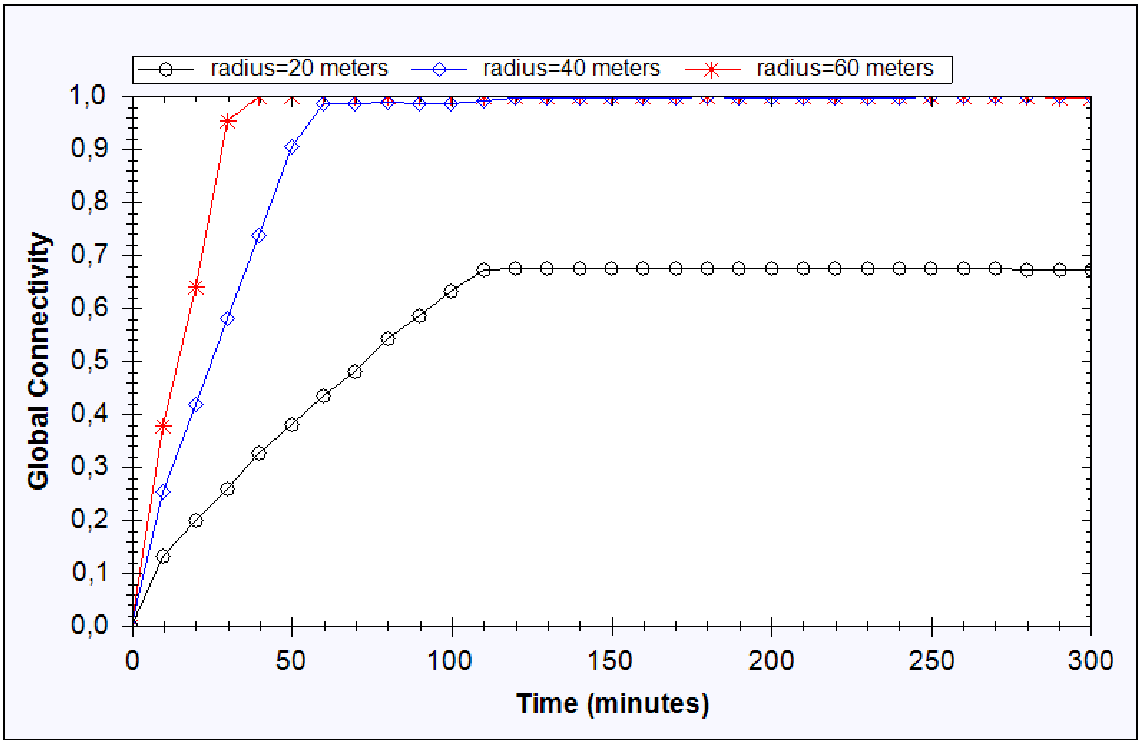


Figure 3.26 Global connectivity versus time for different communication ranges for half-mobile case where *m*=200

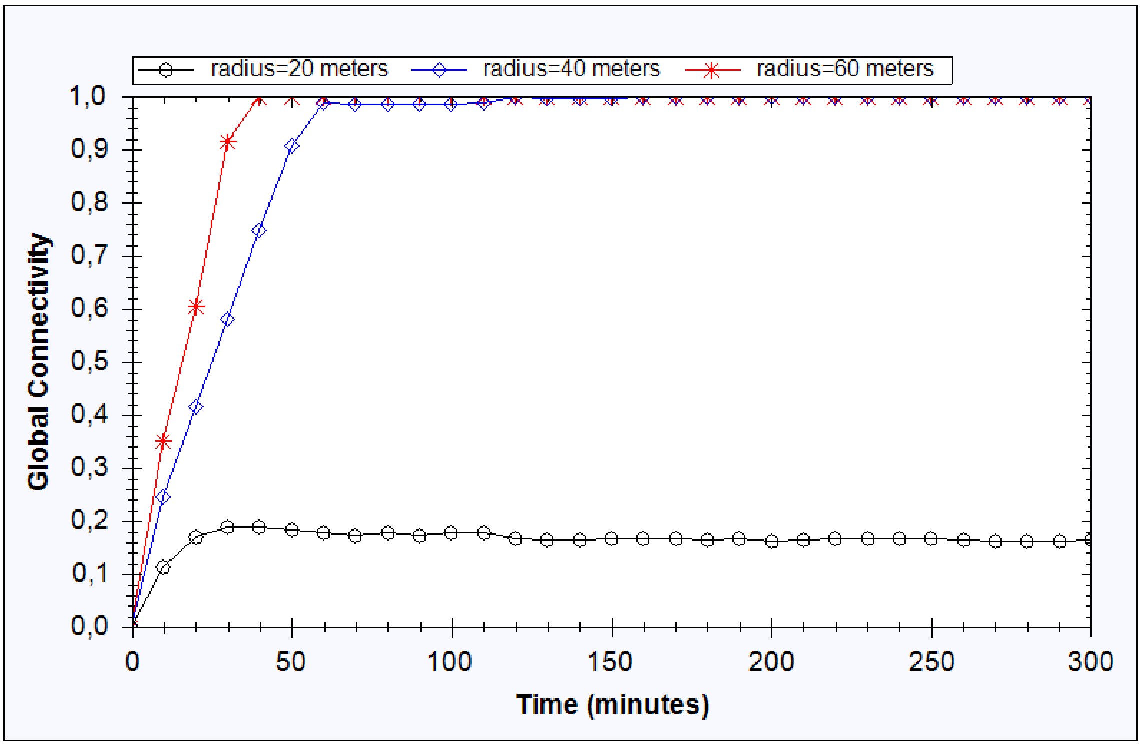


Figure 3.27 Global connectivity versus time for different communication ranges for fully-mobile case where *m*=200

3.3.2.6 Global Connectivity Using Multiple Static Base Stations

Our original scheme uses a mobile BS as a key distribution scheme. However, in this case we use multiple static Base Stations in the environment as key distribution centers and calculate global connectivity. In this case, only the nodes that come to the vicinity of BSs can get keys and there is no guarantee that all the area will be covered. In our simulation we use 25 BSs, 49 BSs and 100 BSs in the environment. With 25 BSs we cover 12.5%, of the sensor field. Similarly with 49 BSs we cover 25% and with 100 BSs we cover 50% of the sensor field. Figure 3.28 and Figure 3.29 show global connectivity values when there are 25, 49 and 100 static BSs in the environment for half mobile and fully mobile cases. Key chain size is kept constant at 200.

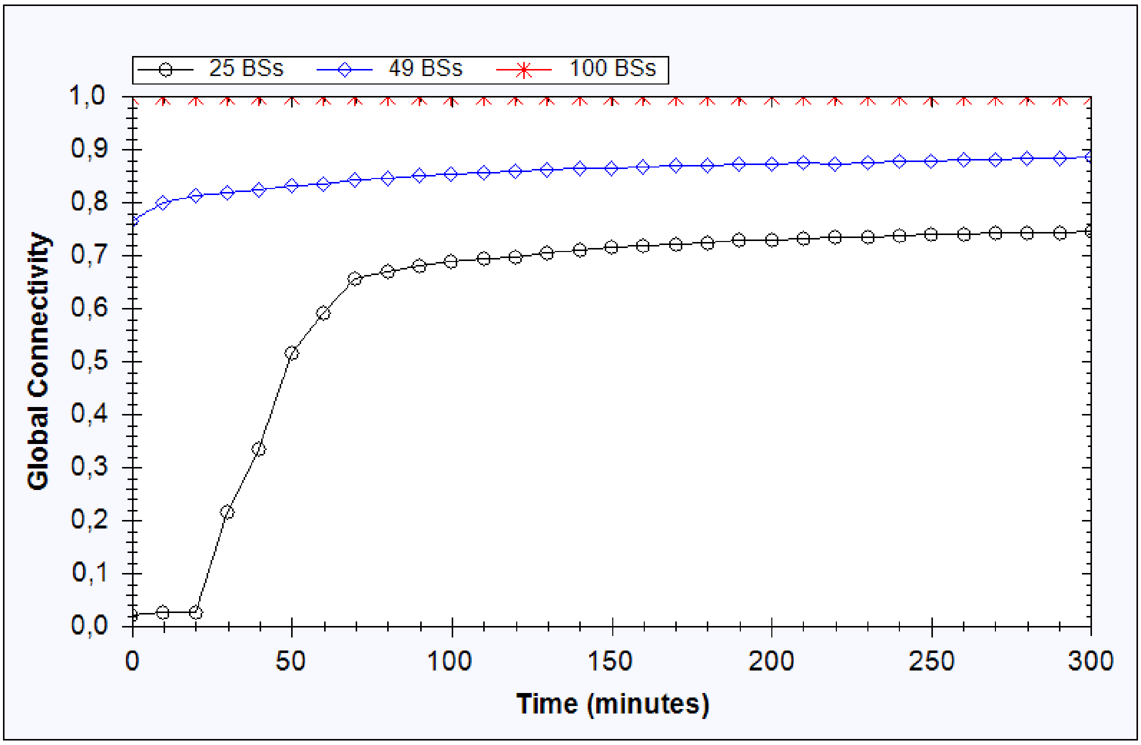


Figure 3.28 Global connectivity versus time using multiple static BSs for half-mobile case where *m*=200

Figure 3.28 shows that as BS number increases in the environment global connectivity gets higher. Global connectivity is always 1 when there are 100 BSs in the environment. The value is around 0.89 for 49 BSs when convergence value is reached, and it is around 0.75 for 25 BSs. When we use 25 BSs, global connectivity is very low at the beginning of the simulation. This is because at first, only a small number of nodes that are around BSs can get keys and form small islands that are disconnected from each other. However as nodes start moving, network gets connected.

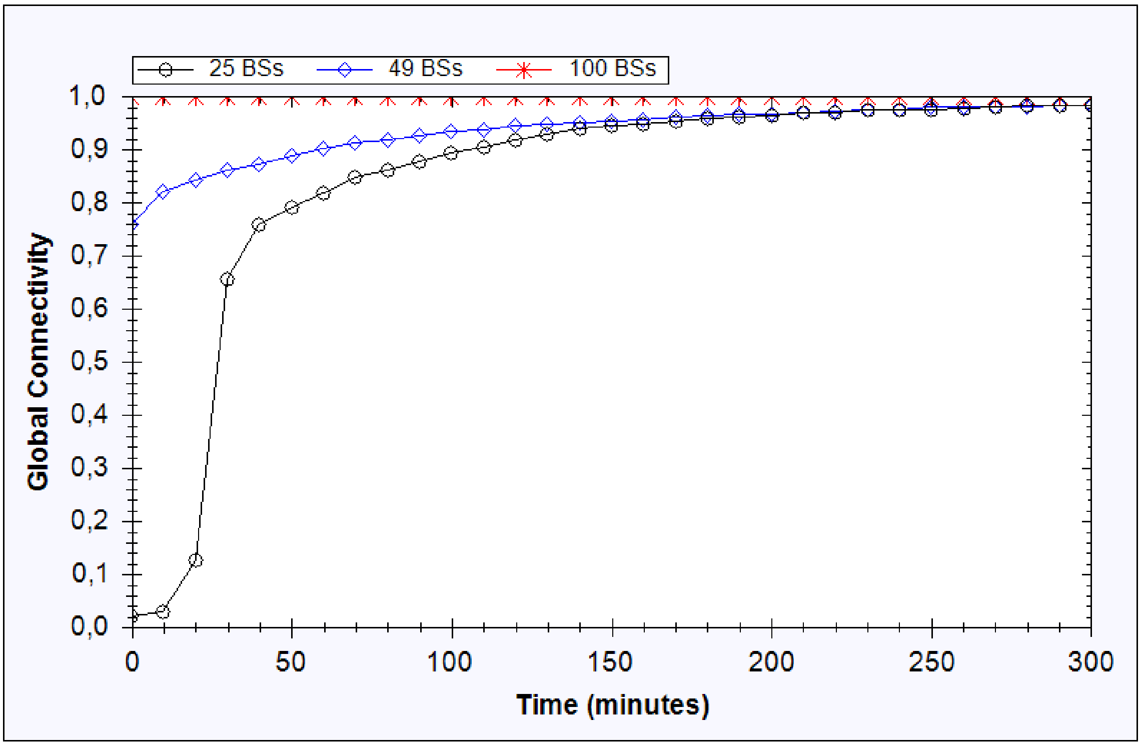


Figure 3.29 Global connectivity versus time using multiple static BSs for fully-mobile case where m=200

The values for fully-mobile case which is shown in Figure 3.29, are higher than half-mobile case. In this figure even 25 static BSs reach global connectivity values very close to 1, because all the nodes are mobile and they can easily come into contact with one of the BSs in the environment when they are traveling.

3.3.3 Resilience

It is possible that an attacker can capture some of the nodes in the network. Since sensor nodes are generally not tamper-proof, an attacker can get access to the key chains of the captured nodes. This way, if the attacker puts the nodes to the network again, it can decrypt the messages sent to and from the captured nodes. Moreover, in some key distribution schemes, it is possible that an attacker can even compromise the links between non-captured nodes as well. Resilience is defined as the ratio of additional compromised communication links to the all communication links in the network [3]. It is a measure showing how much extra links can get compromised by the attacker. When calculating resilience, the nodes that are captured by the attacker and their links are not counted, instead the damage this node capture can bring to other healthy nodes and links are calculated.

In our scheme, a node only keeps its secret key with Base Station and pairwise keys it has with its neighbors. Please note that these pairwise keys are generated and distributed by the BS. A node does not keep pair-wise keys of any other node pairs. Thus, even if an attacker captures a number of nodes and gets access to the keys carried by those nodes, it can only compromise the links of the already captured nodes. However, it cannot compromise any additional links with the help of those keys. This means that our scheme has perfect resiliency against node capture.

Chapter 4

Incorporating Other Key-distribution Schemes into Our Scheme

We explained our key distribution scheme for mobile wireless sensor networks in Chapter 3. Our scheme achieves a good local and global connectivity values and brings perfect resilience against node capture. In our proposed scheme, sensor nodes do not get preloaded with keys except for the secret key they share with Base Station, BS. However, since the key distribution process is done through a mobile Base Station which works as a key distribution center, it takes a while for all the nodes to get pairwise keys for their neighbors and for the network to be connected. This time needed to connect the nodes depends on the speed of BS or the number of BSs in the environment. This situation makes the network unconnected for a while and nodes unable to communicate with each other during this period.

In this chapter, we propose to use other key-distribution schemes together with our scheme in order to solve the problem mentioned above. The idea is to use a key distribution scheme namely Basic Scheme [2] or Du’s Scheme [3], which pre-distributes some keys to the nodes at first and shift to our scheme over time. In this chapter, we explain how we incorporate these two key distribution schemes into our scheme. Section 4.1 explains incorporating Basic Scheme and shows how network performs in terms of local connectivity, global connectivity and resilience against node capture. Section 4.2 uses Du's Scheme in a similar way and shows the performance results.

4.1 Incorporating Basic Scheme into Our Scheme

The random key pre-distribution scheme proposed by Eschenauer and Gligor [2], which we refer to as Basic Scheme, is the first probabilistic key distribution scheme. The idea of the scheme is to preload sensor nodes with keys from a global key pool randomly and let the nodes discover shared keys after deployment and if necessary establish path keys to achieve high network connectivity. In this section, we propose a hybrid scheme which incorporates Basic Scheme into our scheme.

Our proposed scheme has four components; initialization, key distribution, shared-key discovery and update of the key chains. Each step is explained below:

- *Initialization Phase:* This phase covers the initial node configuration and deployment. Before deployment, the nodes are preloaded with keys according to the Basic Scheme. Moreover, each node i is preloaded with a unique pairwise key K_{i-BS}, which it shares with the Base Station. The other configurations with regards to mobility are the same as our original scheme. After configuration, nodes are uniformly deployed to the area.
- *Key Distribution Phase:* When a node senses BS in its communication range key distribution phase takes place. This phase is the same as the original scheme's phase; the protocol is explained in Section 3.2.
- *Shared-key Discovery Phase:* When a node n_i wants to communicate with a neighboring node, n_i first looks at keys it got form BS to see if it has a common pair-wise key with that node. If they have a common pairwise key, they use that key for the communication. If they do not have a common pairwise key, then the n_i tries the keys it got from Basic Scheme and they perform a shared key discovery process as described in original Basic Scheme. Flow of the shared key discovery phase for a single node n_i is described in Figure 4.1.

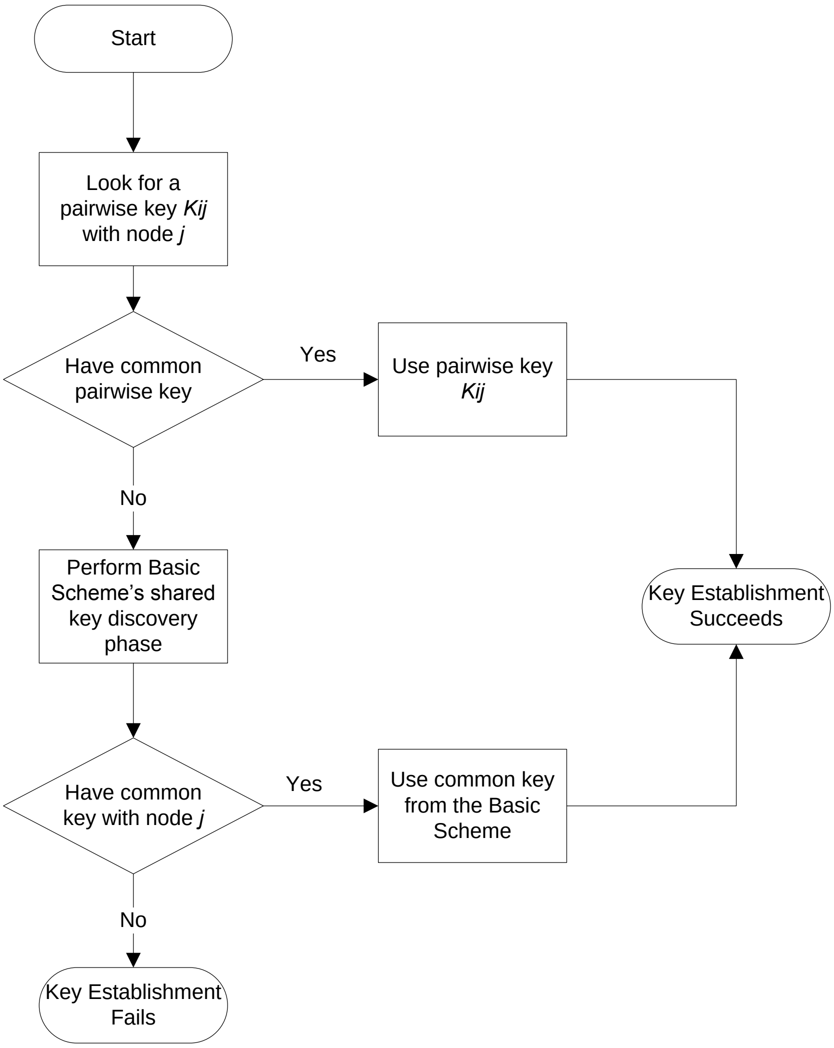

Start
Look for a pairwise key *Kij* with node *j*
Have common pairwise key
Yes
Use pairwise key *Kij*
No
Perform Basic Scheme's shared key discovery phase
Key Establishment Succeeds
Have common key with node *j*
Yes
Use common key from the Basic Scheme
No
Key Establishment Fails

Figure 4.1 Shared key discovery phase using Our Scheme + Basic Scheme

- *Update of the Key Chain:* As mentioned earlier we have a fixed key chain size. Therefore we use a first-in-first-out update mechanism to use the key chain effectively. In this hybrid scheme, once a node meets with the Base Station, BS starts distributing pairwise keys to the node according to our scheme. When a node gets new pair-wise keys from the BS, it randomly selects a key provided by the Basic Scheme from its key ring, deletes that key and adds the new pair-wise key it got from BS to its key chain. This way the size of key chain of the node is kept constant. Using this update mechanism, the predistributed keys of Basic Scheme are gradually replaced with pairwise keys of our scheme. Flow of the update mechanism is shown in Figure 4.2.

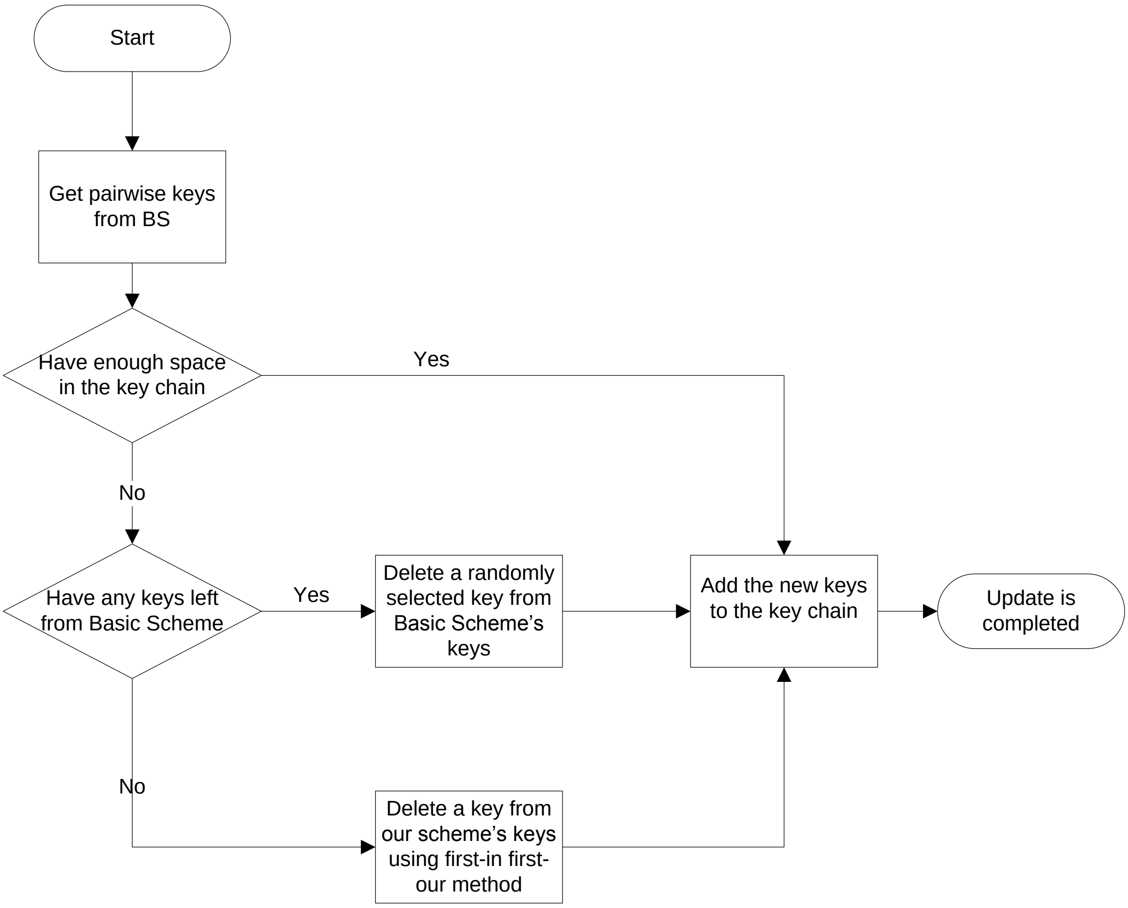

Start
Get pairwise keys from BS
Have enough space in the key chain
Yes
No
Have any keys left from Basic Scheme
Yes
Delete a randomly selected key from Basic Scheme's keys
Add the new keys to the key chain
Update is completed
No
Delete a key from our scheme's keys using first-in first-our method

Figure 4.2 Update of the key chain for Our Scheme + Basic Scheme

To see the performance of the above-described scheme, we conducted simulations for various cases and calculated local connectivity, global connectivity and resilience of the network. The details of these cases and simulation results are explained in the next subsections. The system settings for the Basic Scheme we used for these cases are as follows:

- $|S|$ is the size of key pool. This value is set to 100,000.
- The number of sensor nodes in the network is 10,000.
- The deployment area is 1000m × 1000m.
- The wireless communication range for each node is 40m.

4.1.1 Local Connectivity Performance

To see the local connectivity performance of the proposed scheme incorporated with the Basic Scheme, we conduct simulations for various cases similar to the cases we use for Chapter 3. We use different *m* values, different BS speeds, and multiple Base Stations. The details are explained below.

4.1.1.1 Local Connectivity for Different *m* Values

In order to see how the scheme performs with different key chain size, *m* values, we run simulations for different *m* values and calculated local connectivity. When using this mixed scheme each node is first preloaded with the specified *m* number of keys according to Basic Scheme. After the simulation starts, the keys are slowly replaced by the pairwise keys of our scheme for which the details are explained above. The key chain size *m* we use are *m*=100, *m*=150, *m*=200 and *m*=250. The simulations are conducted for both half-mobile and mobile cases. The results are shown in Figure 4.2 and Figure 4.3. The speed of BS is kept constant at 400 meters/minute.

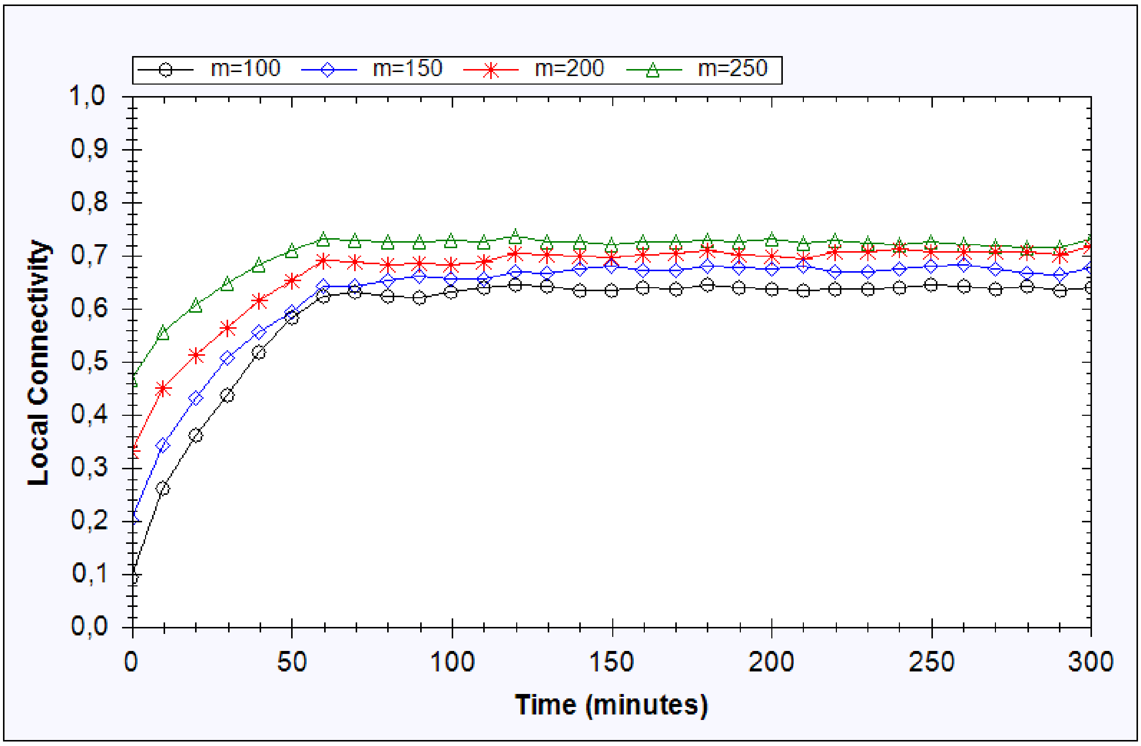


Figure 4.3 Local connectivity versus time for different *m* values using Our Scheme + Basic Scheme for half-mobile case where *BS speed*=400 meters/minute

Figure 4.3 shows that at first the network has some local connectivity. As BS starts its movement, local connectivity gradually increases and finally reaches a convergence value around *time*=60. The initial local connectivity value is 0.09 for *m*=100, 0.2 for *m*=150, 0.33 for *m*=200 and 0.47 for *m*=250. If we compare these results to Figure 3.1, we can see that these values are the local connectivity values of original Basic Scheme. As BS starts moving and distributing pair-wise keys to the nodes, connectivity increases and for all the cases it reaches a convergence value after which the connectivity stays fairly stable. If we compare this figure with Figure 3.10, we can see that the convergence values for both schemes are the same. Eventually this scheme gets the same local connectivity values of our original scheme for all different *m* values. The reason for that is straightforward; as the scheme shifts from Basic Scheme to our scheme so does the connectivity values. Please note that just like Figure 3.10, there is a slight difference between different *m* values' results and it takes some time for the network to reach the convergence value for which the reasons were discussed earlier.

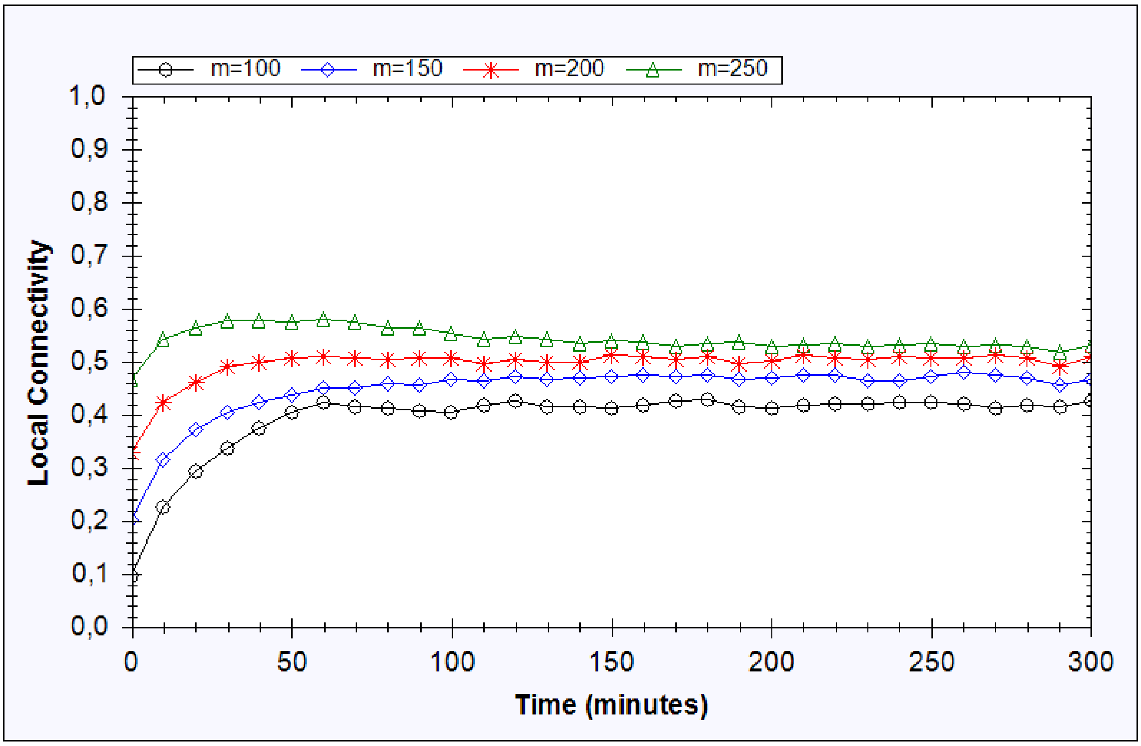


Figure 4.4 Local connectivity versus time for different *m* values using Our Scheme + Basic Scheme for fully-mobile case where *BS speed*=400 meters/minute

Figure 4.4 shows that for each *m* value, the local connectivity starts with the values of Basic Scheme, after which it increases for a while and eventually converges at the values that are the same as our original scheme. The results of the original scheme can be seen in Figure 3.11. Note that for *m*=250, local connectivity first achieves a value a bit higher than the convergence value; however, it decreases to the convergence value after a while. Just like the original scheme, connectivity for fully-mobile case is lower than the connectivity for half-mobile case since nodes move away from each other faster in the fully-mobile case. Also the slight difference in local connectivity for different *m* values can be observed in this figure as well.

4.1.1.2 Local Connectivity for Different BS Speeds

We run simulations to see the performance of the scheme for difference BS speeds. The BS speeds we use for our simulations are; *BS speed*=200 meters/minute, *BS speed*=400 meters/minute and *BS speed*=600 meters/minute. The key chain size *m* is kept constant at 200. The simulations are done for both half-mobile and fully-mobile cases. The results are shown in Figure 4.5 and 4.6.

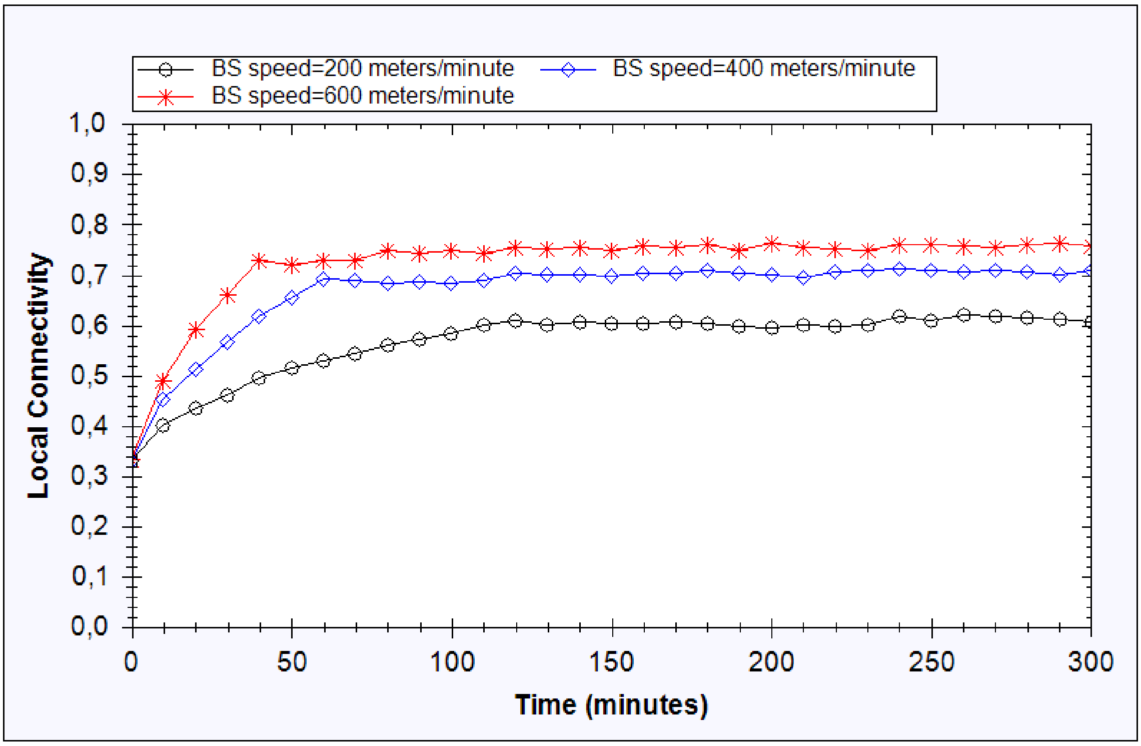


Figure 4.5 Local connectivity versus time for different BS speeds using Our Scheme + Basic Scheme for half-mobile case where *BS speed*=400 meters/minute

As it can be seen in Figure 4.5 for all BS speeds, local connectivity starts from 0.33 and slowly reaches the convergence value of our scheme. The final convergence values are almost the same as original values; for *BS speed*=600, local connectivity is around 0.77, for *BS speed*=400 it is around 0.71 and for *BS speed*=200 local connectivity is around 0.60. Please note that the convergence value is reached at *time*=40 for *BS speed*=600, at *time*=60 for *BS speed*=400 and at *time*=120 for *BS speed*=200. As the speed increases BS completes its one round in the area faster and therefore convergence value is reached sooner.

Figure 4.6 shows the result for fully-mobile case. Again, the connectivity starts with 0.33 and reaches our scheme's convergence values for each speed. Please note that for *BS speed*=200, local connectivity value first reaches 0.4, a value which is little higher than the convergence value. However, after a while it decreases and stabilizes around 0.40. It can be seen that, like the half-mobile case, the time needed to reach convergence value changes for different BS speeds, since it depends on the time BS completes its one round of movement in the area.

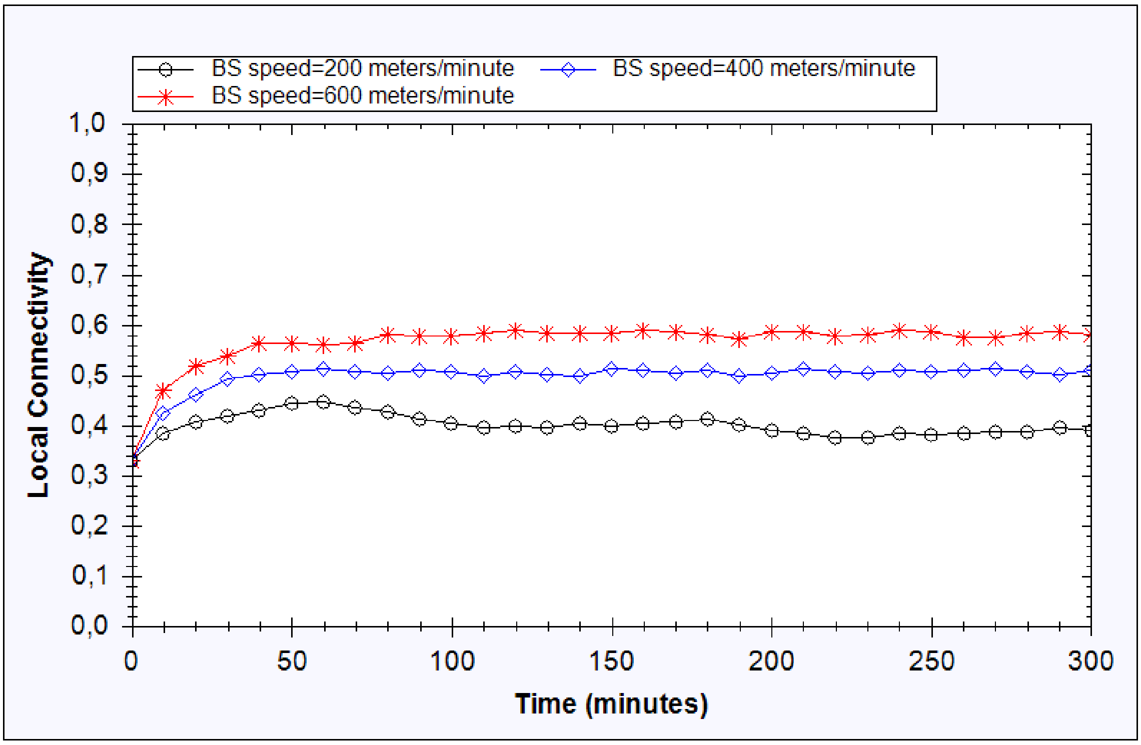


Figure 4.6 Local connectivity versus time for different BS speeds using Our Scheme + Basic Scheme for fully-mobile case where *BS speed*=400 meters/minute

4.1.1.3 Local Connectivity Using Multiple Base Stations

We also conduct simulations using multiple BSs for this proposed scheme. When we use two BS in the environment, one of them moves and operates on the upper half of the network and the other moves and operates on the lower half. The movements of the BSs are done as explained in Figure 3.12. The simulations are conducted for both half-mobile and mobile case. The speeds of BSs are kept constant at 400 meters/minutes and the key chain size *m* is kept constant at 200. Figure 4.7 and 4.8 show the results of the simulation for half-mobile and mobile cases.

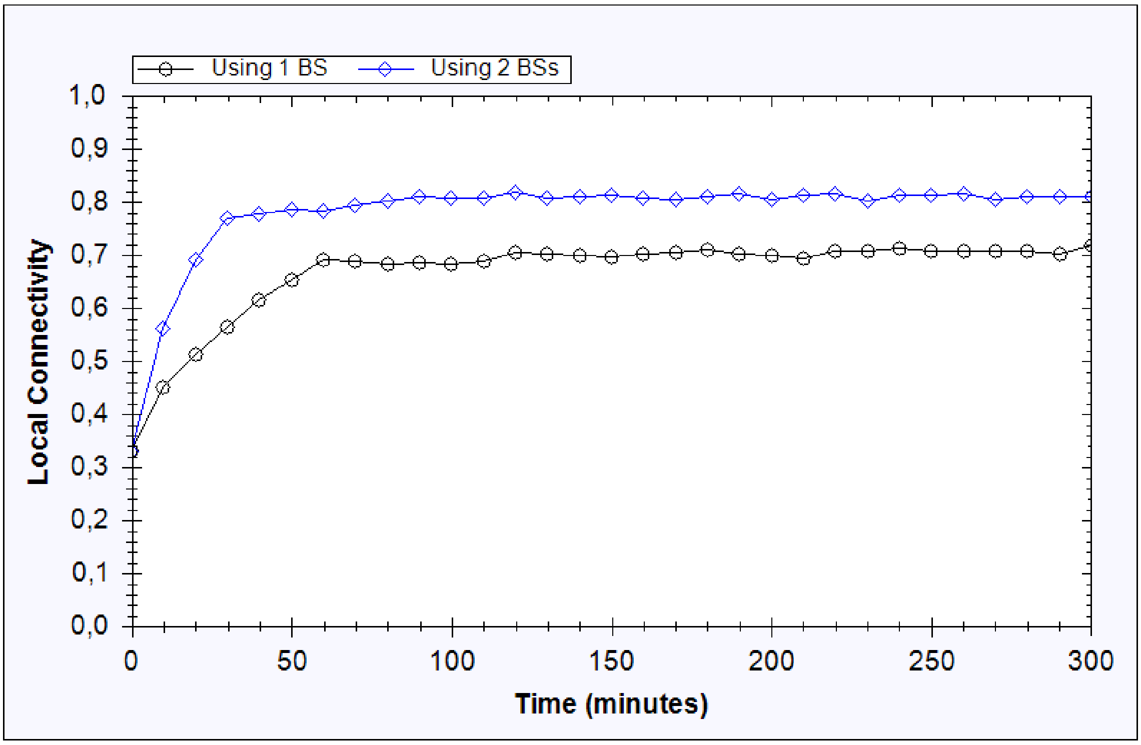


Figure 4.7 Local connectivity versus time using multiple BSs using Our Scheme + Basic Scheme for half-mobile case where *BS speed*=400 meters/minute

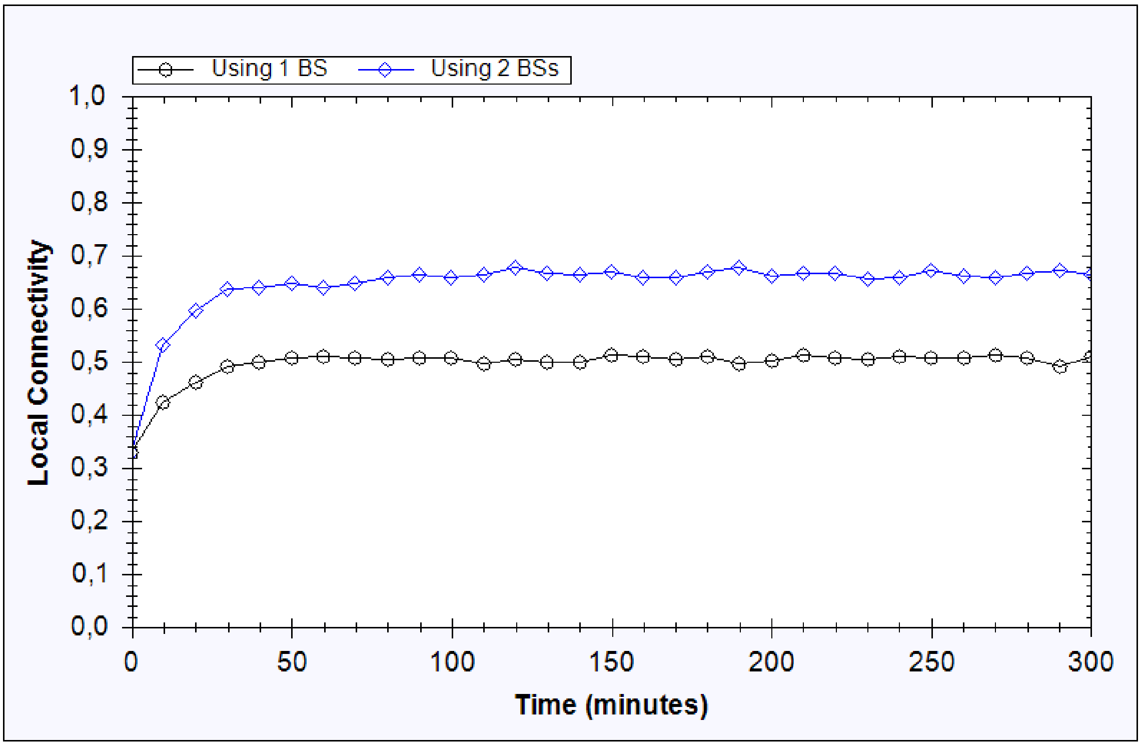


Figure 4.8 Local connectivity versus time using multiple BSs using Our Scheme + Basic Scheme for fully-mobile case where *BS speed*=400 meters/minute

Figure 4.7 and 4.8 shows that local connectivity starts with Basic Scheme's value and reaches its convergence value of our scheme. Using multiple BSs increases the local connectivity as the nodes get their key chains updated more frequently when there are two BSs operating in the field. Also note that when there are two BS in the environment the time needed to achieve the convergence value decreases since the simulation area gets scanned faster in this case.

4.1.2 Global Connectivity Performance

In order to see how our proposed scheme performs in terms of global connectivity we conduct simulations for various cases and calculate global connectivity values over time. The cases and global connectivity results are explained in the following sub-sections.

4.1.2.1 Global Connectivity for Different *m* Values

We run simulations using different *m* values of 100, 150, 200 and 250 and calculated global connectivity over time. Since we use Basic Scheme at the beginning, the network can achieve very high global connectivity values even at the beginning. Table 4.1 shows the global connectivity values for different *m* values at *time*=0 and *time*=300.

Table 4.1 Global connectivity values for different *m* values at *time*=0 and *time*=300 using Our Scheme + Basic Scheme

Key Chain Size, *m*	**Time**	**Half-Mobile**	**Fully-Mobile**
m=100	0	0.9872	0.9879
	300	0.9953	0.9965
m=150	0	0.9998	0.9998
	300	0.9985	0.9979
m=200	0	1	1
	300	0.9988	0.9987
m=250	0	1	1
	300	0.9987	0.9993

Table 4.1 shows that the proposed scheme achieves a very high connectivity even at the beginning of the simulations. This means the nodes can communicate with each other at the earlier phases of the deployment as well. The values at *time*=0 are the values of Basic Scheme. As the keys gets replaced by pair-wise schemes the scheme shifts from Basic Scheme towards our scheme. If we compare the result of Table 4.1 with that of Table 3.1, we can see that the values at *time*=300 are very close to each other for both tables. This shows that just like local connectivity, global connectivity values also converge to the results of our scheme by time. It can also be seen that as *m* increases global connectivity also increases and fully-mobile case generally has a slightly higher global connectivity value compared to half-mobile case.

4.1.2.2 Global Connectivity for Different BS Speeds

We conduct simulations to calculate global connectivity values using different BS speeds as well. The results of the simulations for half-mobile and fully-mobile cases are shown in Table 4.2

Table 4.2 Global connectivity values using different BS speeds at *time*=0 and *time*=300 for *m*=200 using Our Scheme + Basic Scheme

BS speed	Time	Half-mobile	Fully Mobile
BS speed=200 meters/minutes	0	1	1
	300	0.9978	0.9971
BS speed=400 meters/minutes	0	1	1
	300	0.9988	0.9987
BS speed=600 meters/minutes	0	1	1
	300	0.9989	0.9996

Table 4.2 also shows that the scheme starts with Basic Scheme and slowly gets close to our scheme and shows global connectivity values close to our original scheme at the end of the simulation, which was shown in Table 3.2. The table also shows that as the speed of BS increases, there is a slight increase in global connectivity as well. This is because as BS meets with the nodes more frequently it can distribute keys to the nodes more frequently and keep the network connected.

4.1.2.3 Global Connectivity Using Multiple Base Stations

We also run simulations using multiple base stations and calculated global connectivity over time. Table 4.3 shows the global connectivity values for both half-mobile and fully-mobile cases using one BS versus using multiple BSs. The table shows that using two BSs gets a higher global connectivity value even if the difference is very little. Also note that at *time*=0, global connectivity values are that of original Basic Scheme. At the end of the simulations there is a slight decrease in global connectivity values as they get close to our original scheme's values. This decrease is due to the mobile nature of the nodes. In a mobile network, it is possible that some of the nodes move to geographic locations which are far from other nodes and get isolated from the rest of the network, thus decreasing global connectivity.

Table 4.3 Global connectivity for half-mobile and fully-mobile cases using multiple BSs using Our Scheme + Basic Scheme

BS number	Time	Half-mobile	Fully Mobile
Using 1 BS	0	1	1
	300	0.9988	0.9987
Using 2 BS	0	1	1
	300	0.9998	0.9998

4.1.3 Resilience

As explained earlier, resilience is the ratio of additional compromised communication links to the all communication links in the network when an attacker captures certain amount of nodes. In Chapter 3, we explained that our scheme has perfect resilience against node capture since a node does not carry a key which is used between any other non-captured node pair. However, in this section we use Basic Scheme at the beginning of the node deployment and slowly change to our scheme over time. In Basic Scheme, a node can carry a key which can also exist in another node's key chain since keys are loaded randomly to nodes. Therefore, it is possible that an attacker can make use of the keys it acquired through captured nodes to compromise links between non-captured nodes as well. This means there is some vulnerability against node capture in terms of resilience for Basic Scheme. Since our scheme uses Basic Scheme, our scheme is also affected by that vulnerability.

To calculate the resilience against node capture, we use two different attack models. In the first model, we have an attacker who captures a certain amount of nodes at the very beginning of the simulation. This model is used to see what happens in the worst case. In the second model, we have a more likely to occur attack model. In this model, an attacker captures nodes gradually. We run simulation for these two attack models. The results are explained in the following sub-sections.

4.1.3.1 Worst Case Attack Scenario

In this model, an attacker captures a number of nodes randomly in the environment and gets access to their keys. The node captures are assumed to happen all at the same time, at the beginning of the simulations. The attacker cannot attack Base Station and cut its communication with the nodes. To see the performance we first calculate additional compromised link ration of the network over time. Moreover, we also look at the ratio of total compromised links to all communication links, meaning in this second calculation we took captured nodes into account as well. For the number of captured nodes, we used 200. We kept *m* constant at 200 and *BS speed* constant at 400 meters/minutes. Figure 4.9 and 4.10 shows the additional compromised links ratio and total compromised links ratio for both half-mobile and fully- mobile cases.

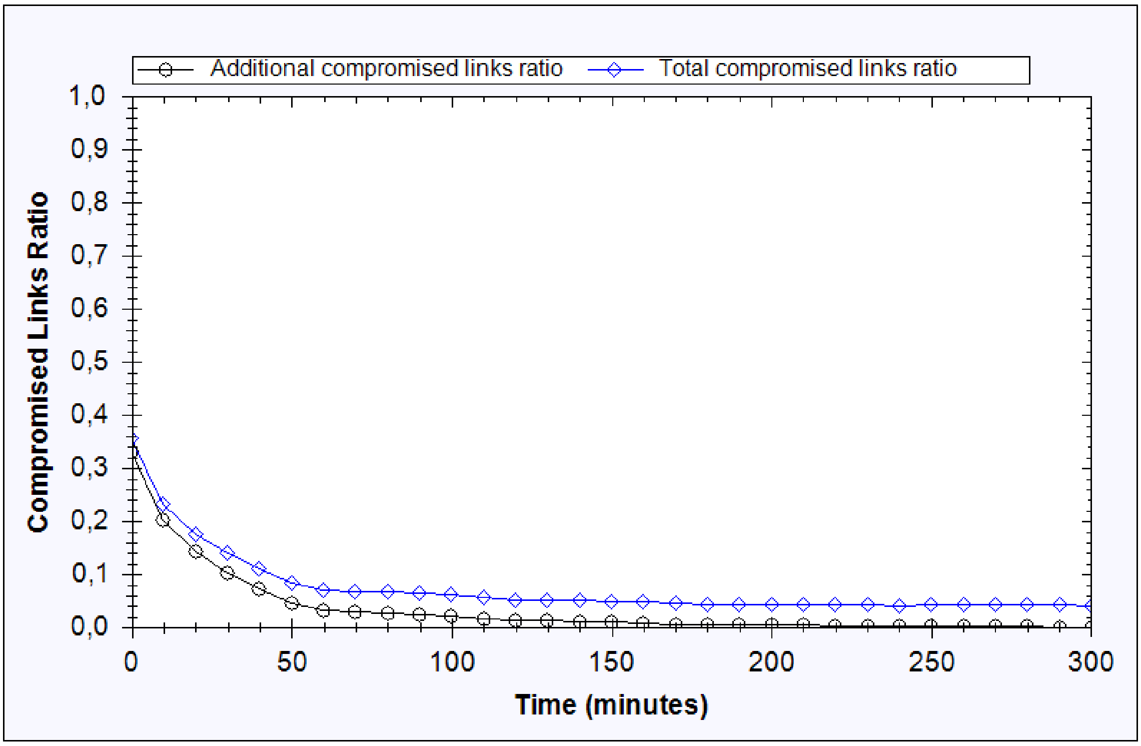


Figure 4.9 Additional and total compromised links ratio for captured node count=200 for half-mobile case using Our Scheme + Basic Scheme for Worst Case Attack Scenario

Figure 4.9 shows that additional compromised links ratio at *time*=0 is around 0.33 and it slowly decreases after that. As Basic Scheme is slowly replaced by our scheme, this ratio decreases and eventually becomes 0. This is to be expected, since as our scheme starts to dominate, nodes delete their old keys from the Basic Scheme, which makes the attacker’s keys which he/she got at the beginning useless. Total compromised links ratio is littler higher then additional compromised links ratio, because in this case all the links, not just the additional compromised links are taken into account. Nevertheless, it also follows the same pattern and decreases by time stabilizing around 0.04 which is basically the ratio of captured nodes’ communication links to all links in the network.

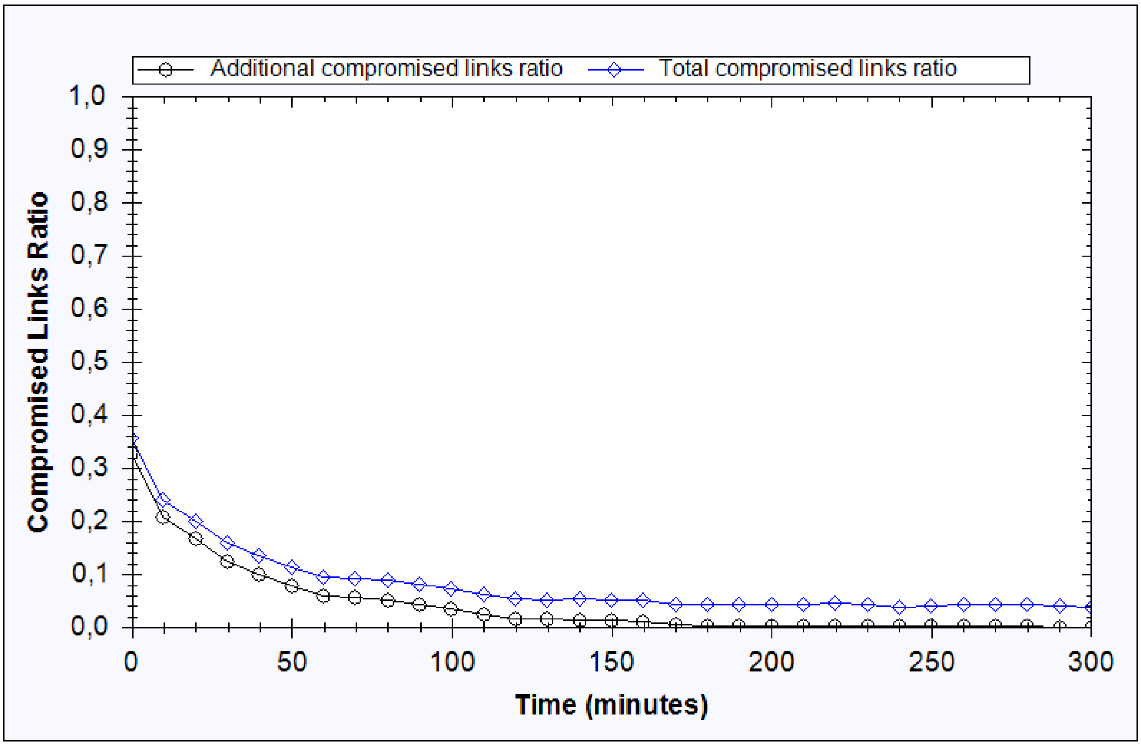


Figure 4.10 Additional and total compromised links ratio for captured node count=200 for fully mobile case using Our Scheme + Basic Scheme for worst case attack scenario

Fully-mobile case shows similar results to and almost the same values as half-mobile case. The reason is that rather than the nodes' mobility, the improvement in resilience and the decrease in total compromised links ratio depend on the BS's movement, how frequent a node meets BS, how frequent it gets new keys and deletes Basic Scheme's keys. In this figure as well, we can see that additional compromised links ratio and total compromised links ratio gets smaller by time, additional compromised links ratio getting 0, and total compromised links getting close to 0.04.

4.1.3.2 Typical Attack Scenario

In this attack model, attacker captures the nodes gradually over time. As mentioned earlier this is a more likely to occur attack scenario. In this scenario we let the attacker capture one node per minute and the total amount of captured nodes are set to be 200. We kept *m* constant at 200 and BS speed constant at 400 meters/minute. Figure 4.11 and 4.12 show the compromised links ratio for half-mobile and fully-mobile cases.

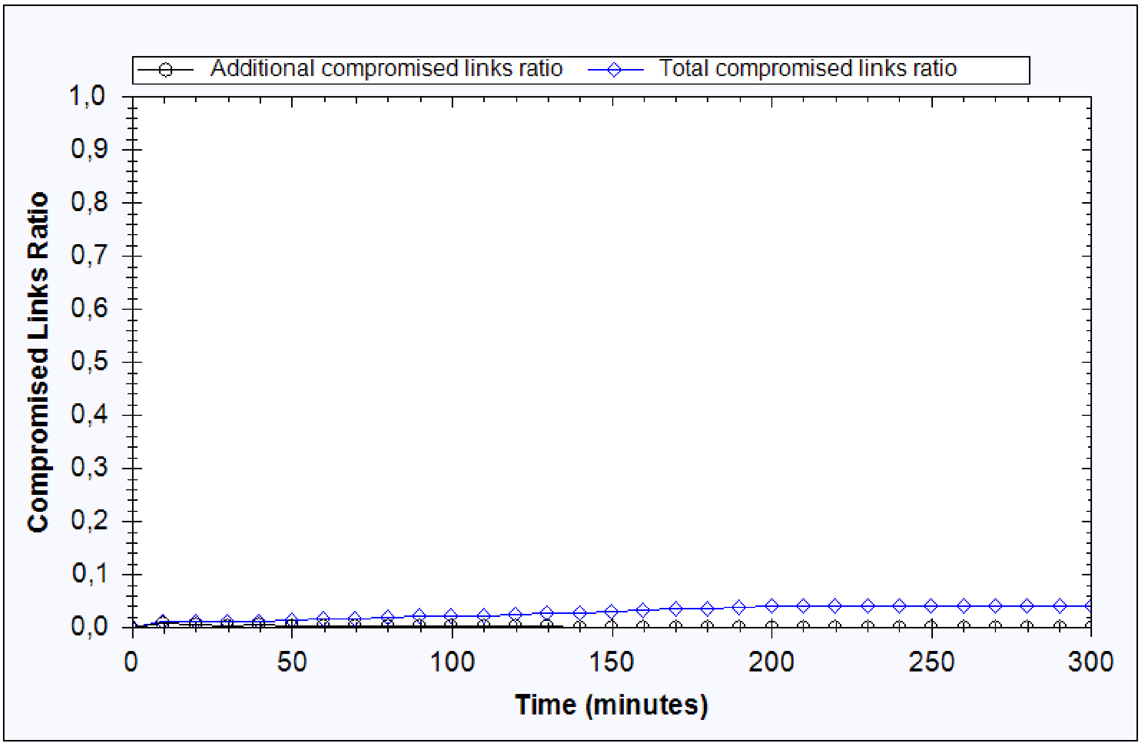


Figure 4.11 Additional and total compromised links ratio for captured node count=200 for half mobile case using Our Scheme + Basic Scheme for typical attack scenario

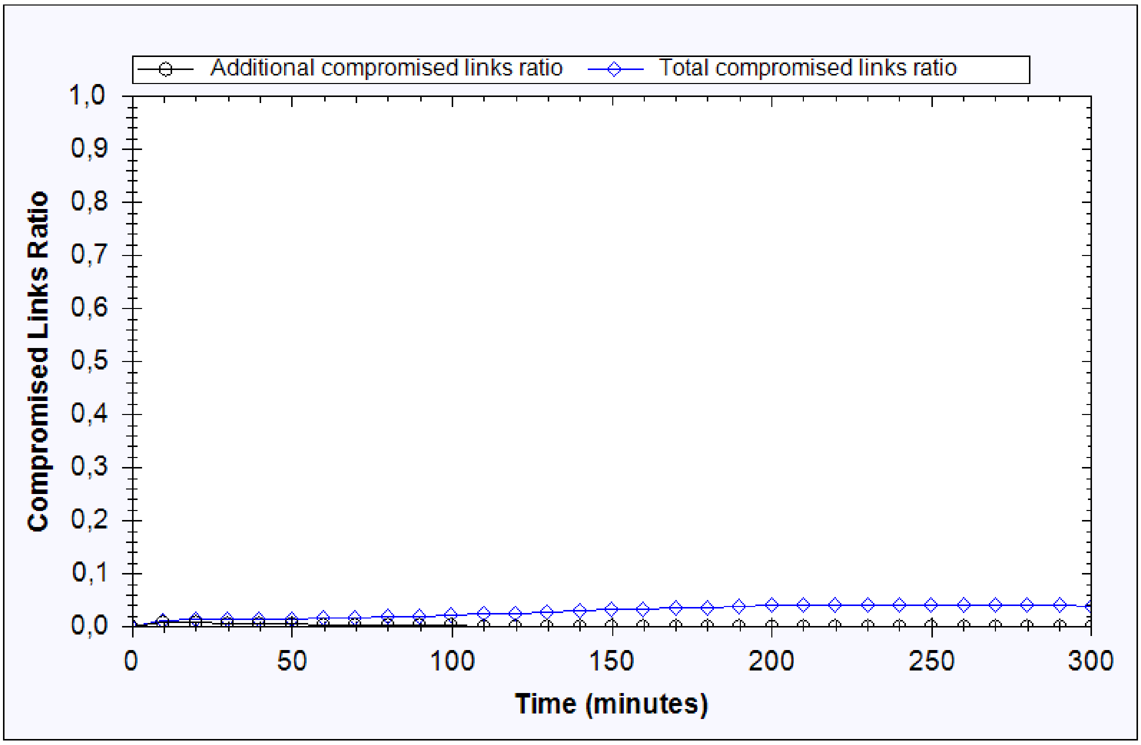


Figure 4.12 Additional and total compromised links ratio for captured node count=200 for fully-mobile case using Our Scheme + Basic Scheme for typical attack scenario

It can be seen in Figure 4.11 and 4.12 that resilience of the scheme for this attack model is very good. Additional compromised links ratio increases a little at the beginning of the simulation; however it decreases and becomes close to 0 as time passes. This is because as time passes, the keys of Basic Scheme are being deleted from the key chain. So even if an attacker captures nodes, it has very little help to him/her in compromising any additional links. Also note that total compromised links ratio converges to 0.04 which is the ratio of the total compromised links to all the links in the network.

4.2 Incorporating Du's Scheme into Our Scheme

Du et al. [3] proposes a probabilistic key distribution scheme using deployment information of the nodes to achieve high local and global connectivity of the network. The idea they use is to divide the nodes into groups and deploy them to the area which is divided into zones using Gaussian distribution. The key pool is also divided into groups and nodes deployed to the same zone get keys from their specified key pool so that they have a high probability of sharing same keys. They also get a number of keys from their neighboring zones' key pools, so that the network connectivity in general is also kept high. In this part, we propose hybrid scheme which incorporates Du's Scheme into our scheme.

Our proposed scheme has four components, namely; initialization phase, key distribution phase, shared-key discovery phase and update of the key chains. The details of the each component are explained below.

- *Initialization Phase:* Initial node configuration and deployment of the nodes are done in this phase. Before deployment, the nodes are preloaded with keys according to the Du's Scheme. Additionally, each node i is preloaded with a unique pairwise key K_{i-BS}, which it shares with the Base Station. The other configurations regarding node mobility are the same as our original scheme. After configuration, nodes are deployed to the area using Gaussian distribution according to Du's Scheme.

- *Key Distribution Phase:* This phase takes place when a node meets with BS. Key distribution phase is the same as the original scheme's key distribution phase; the protocol is explained in Section 3.2.
- *Shared-key Discovery Phase:* When a node n_i wants to communicate with a neighboring node n_j, n_i first checks its key chain to see if it has a common pair-wise key with that node perivously distributed by BS. If they have a common pairwise key, they use that key for the communication. If they do not have a common pairwise key, then the n_i tries the keys it got from Du's Scheme and they perform a shared key discovery process as described in the original Du's Scheme. Flow of the shared key discovery phase for a single node n_i is described in Figure 4.13.

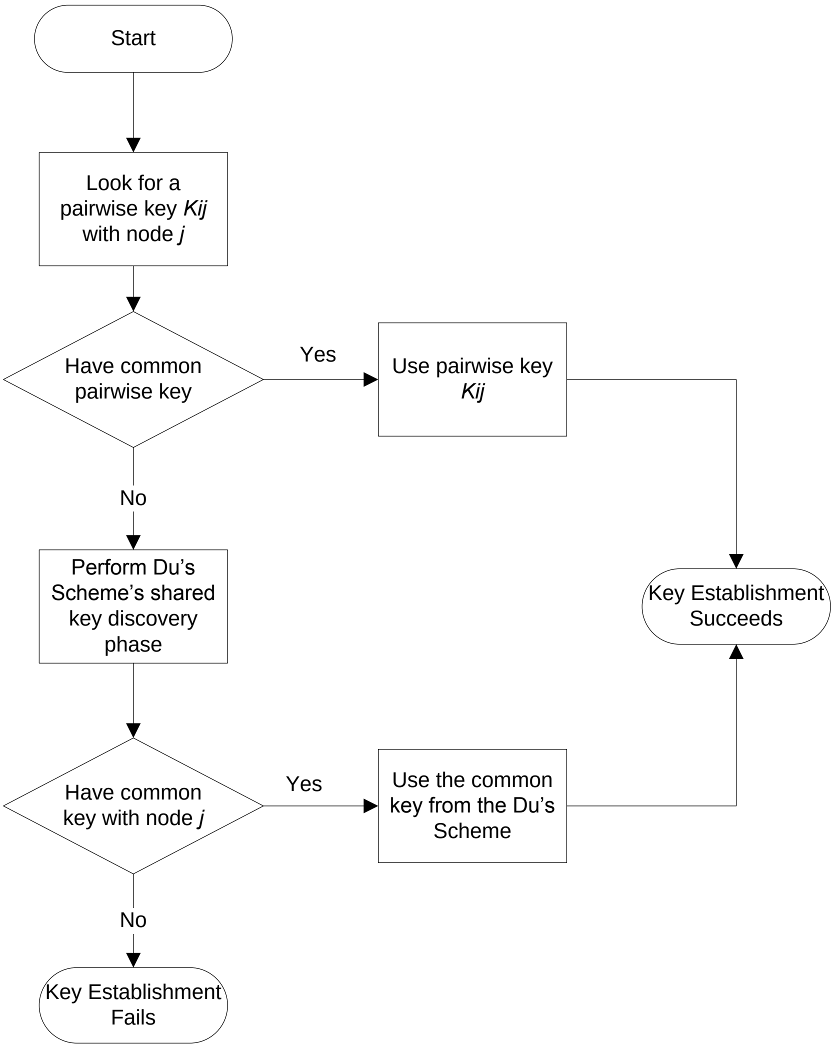

Start
Look for a pairwise key *Kij* with node *j*
Have common pairwise key
Yes
Use pairwise key *Kij*
No
Perform Du's Scheme's shared key discovery phase
Key Establishment Succeeds
Have common key with node *j*
Yes
Use the common key from the Du's Scheme
No
Key Establishment Fails

Figure 4.13 Shared key discovery phase using Our Scheme + Du's Scheme

- *Update of the Key Chain:* Since we have a fixed key chain size, we use a first-in-first-out update mechanism to use the key chain effectively. In this hybrid scheme, when a node meets with the Base Station, BS starts sending pairwise keys to the node according to our original scheme. When a node gets new pair-wise keys from the BS, it randomly selects a key provided by Du's Scheme from its key ring, deletes that key and adds the new pair-wise key it got from BS to its key chain. By this mechanism, the size of key chain of the node is kept constant. Using this update mechanism, the predistributed keys of Du's Scheme are gradually replaced with pairwise keys of our scheme. Flow of the update mechanism is shown in Figure 4.14.

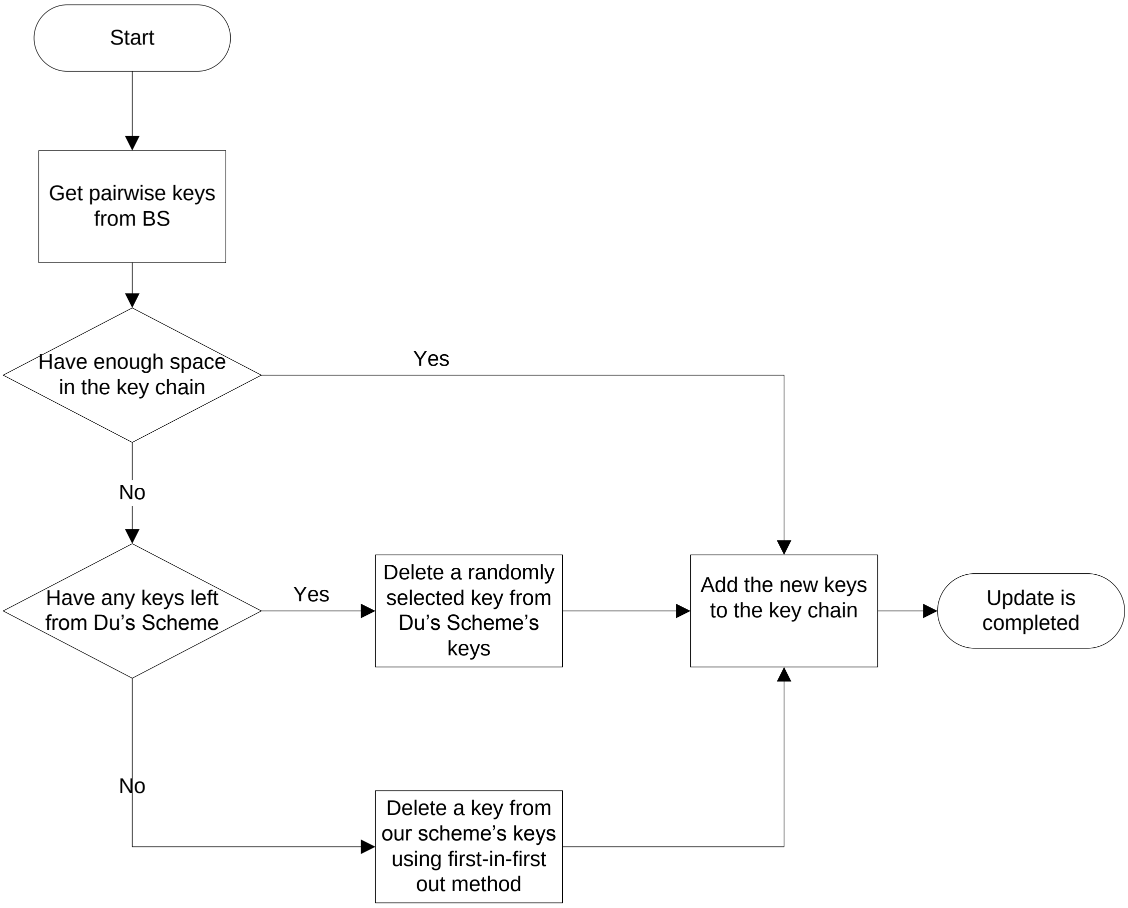

Start
Get pairwise keys from BS
Have enough space in the key chain
Yes
No
Have any keys left from Du's Scheme
Yes
Delete a randomly selected key from Du's Scheme's keys
Add the new keys to the key chain
Update is completed
No
Delete a key from our scheme's keys using first-in-first out method

Figure 4.14 Update of the key chain for Our Scheme + Du's Scheme

In order to see how this scheme works we conduct simulations and calculate local connectivity, global connectivity and resilience of the network against node capture. The results of these simulations are explained in the following subsections. System settings we use for these simulations are as follows:

- $|S|$ is the size of key pool. This value is set to 100,000.
- The number of sensor nodes in the network is 10,000.
- The deployment area is 1000m × 1000m.
- The wireless communication range for each node is 40m.
- The area is divided into a grid of size 100, with each grid cell of size 100m x 100m.
- The center of each grid is the deployment point.

4.2.1 Local Connectivity Performance

In order to see the local connectivity performance of the proposed scheme we run simulations for different cases like different *m* values, different BS speeds and using multiple BSs. The results of these cases are explained below.

4.2.1.1 Local Connectivity for Different *m* Values

Similar to previously proposed schemes, we run simulations for different *m* values for this scheme as well. The values we use are m=100, m=150, m=200 and m=250. The BS speed is kept constant at 400 meters/minutes. The simulations are conducted for both half-mobile and fully mobile cases. Figure 4.15 and 4.16 show the results for both cases.

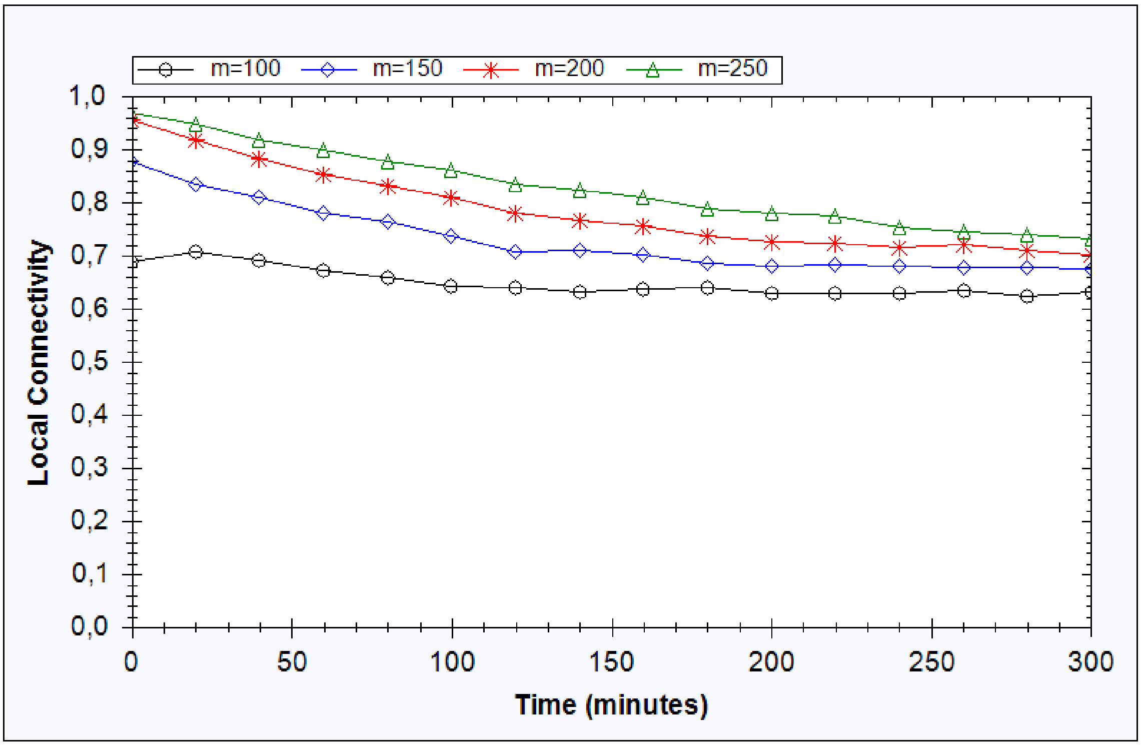


Figure 4.15 Local connectivity versus time for different *m* values using Our Scheme + Du's Scheme for half-mobile case where *BS speed*=400 meters/minute

Figure 4.15 shows that for each *m* value, local connectivity starts from its original value and by time it decreases to our original scheme's convergence values. For *m*=100, local connectivity starts at 0.69, for *m*=150, it starts at 0.88, for *m*=20,0 it starts at 0.95 and for *m*=250, it starts at around 0.97. Towards the end of the simulation, local connectivity for *m*=100 is at 0.64, for *m*=200 it is around 0.68 for *m*=200 it is around 0.70, and for *m*=250 it is at around 0.73. These final values are the same values we got with our original scheme. This is to be expected since keys from Du's Scheme are replaced by our pairwise keys and the scheme shifts to our scheme over time. Note that for *m*=100, at first there is a slight increase in local connectivity but it decreases to the convergence value later.

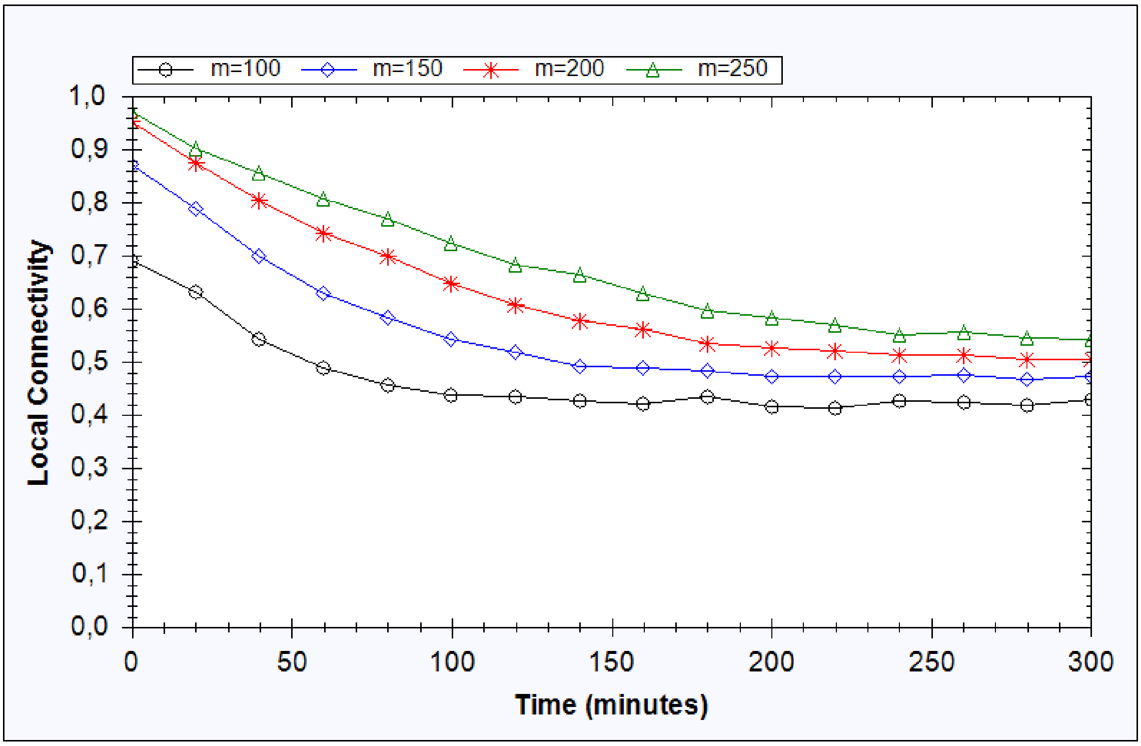


Figure 4.16 Local connectivity versus time for different *m* values using Our Scheme + Du's Scheme for fully-mobile case where *BS speed*=400 meters/minute

Figure 4.16 shows the results for fully-mobile case. It is seen that for all different values of *m*, local connectivity starts from Du's Scheme's values and slowly descends to our original scheme over time. The values for local connectivity for *m*=100 is 0.43, for *m*=150 it is 0.45 for *m*=200 it is 0.52, and for *m*=250 it is around 0.54.

4.2.1.2 Local Connectivity for Different BS Speeds

We also run simulations using different BS speeds. For BS speeds, we use *BS speed*=200 meters/minute, *BS speed*=400 meters/minute and *BS speed*=600 meters/minute. The key chain size is kept constant at 200. The simulations are conducted for both half-mobile and fully-mobile cases. The results are shown in Figure 4.17 and 4.18.

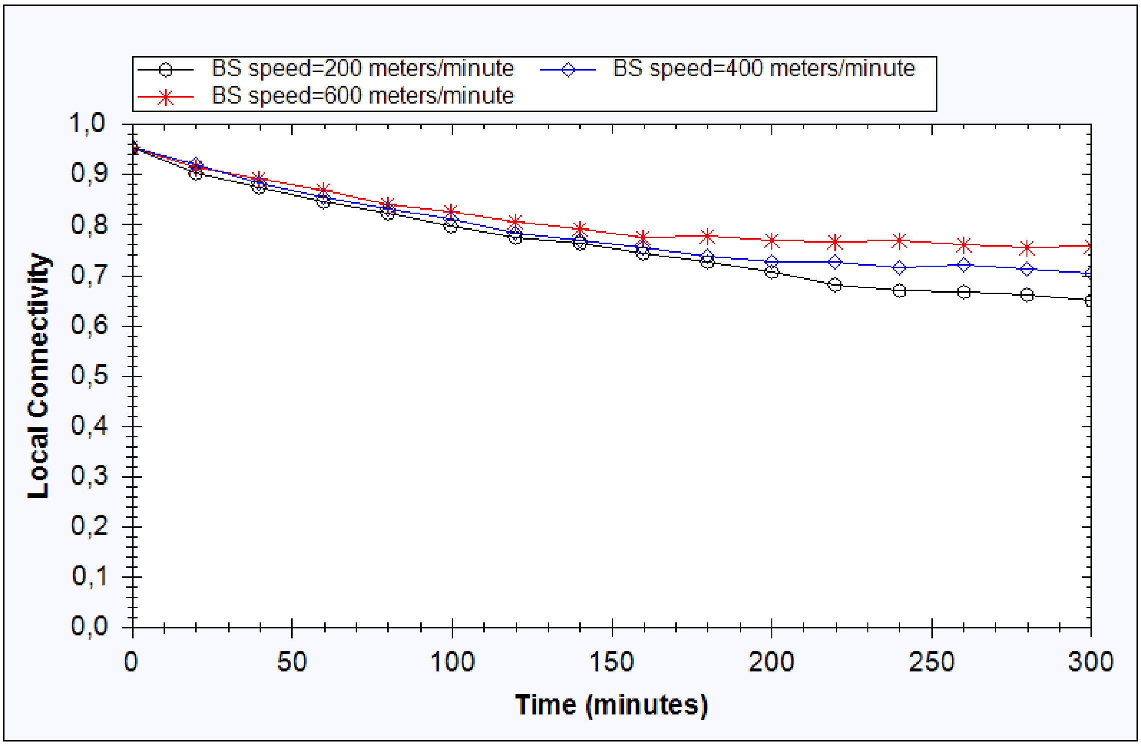


Figure 4.17 Local connectivity versus time for different BS speeds using Our Scheme + Du's Scheme for half-mobile case where *BS speed*=400 meters/minute

Figure 4.17 shows that for all three BS speeds, local connectivity starts from 0.95 and decreases to their respective convergence value over time. The convergence values are the same as our original scheme. Note that it takes the most time for the slowest case to converge to its local connectivity value. This is because the BS in this case meets with the nodes less frequently compared to faster cases and this results in the need for a long time to converge.

Figure 4.18 also shows similar results for half-mobile case. Local connectivity values are of course lower than half-mobile case for which the reasons are explained earlier. For this case as well, the time needed for the network to reach its convergence value is longer for the slowest case.

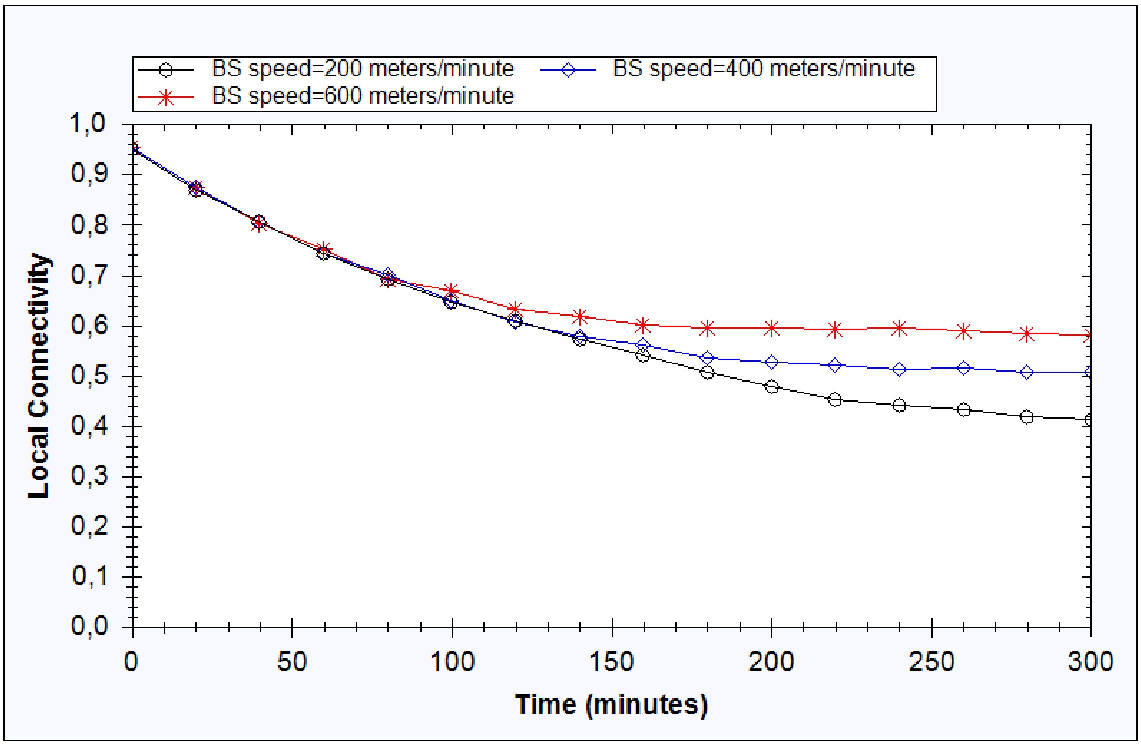


Figure 4.18 Local connectivity versus time for different BS speeds using Our Scheme + Du’s Scheme for fully-mobile case where *BS speed*=400 meters/minute

4.2.1.3 Local Connectivity Using Multiple Base Stations

We run simulations using multiple Base Stations in the environment for this scheme as well. The movements of BSs in the environment are the same as previous schemes. BS speed is kept constant at 400 meters/minute and key chain size *m* is kept constant at 200. The local connectivity values for both half-mobile and fully-mobile case can be seen in Figure 4.19 and 4.20. The figures show that again, nodes start with local connectivity values of Du’s Scheme and over time they converge to our original scheme’s respective values. Using two BSs in the environment keeps the local connectivity at a higher value since nodes can update their key chains more frequently in that case.

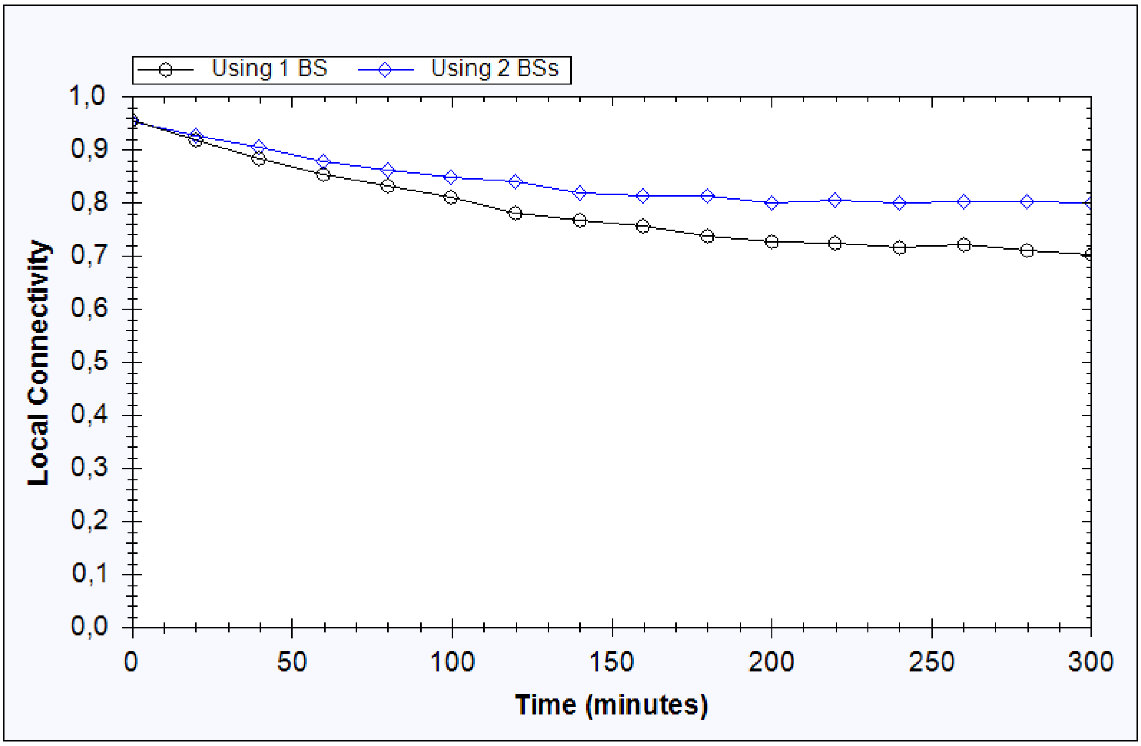


Figure 4.19 Local connectivity versus time using multiple BSs using Our Scheme + Du's Scheme for half-mobile case where *BS speed*=400 meters/minute

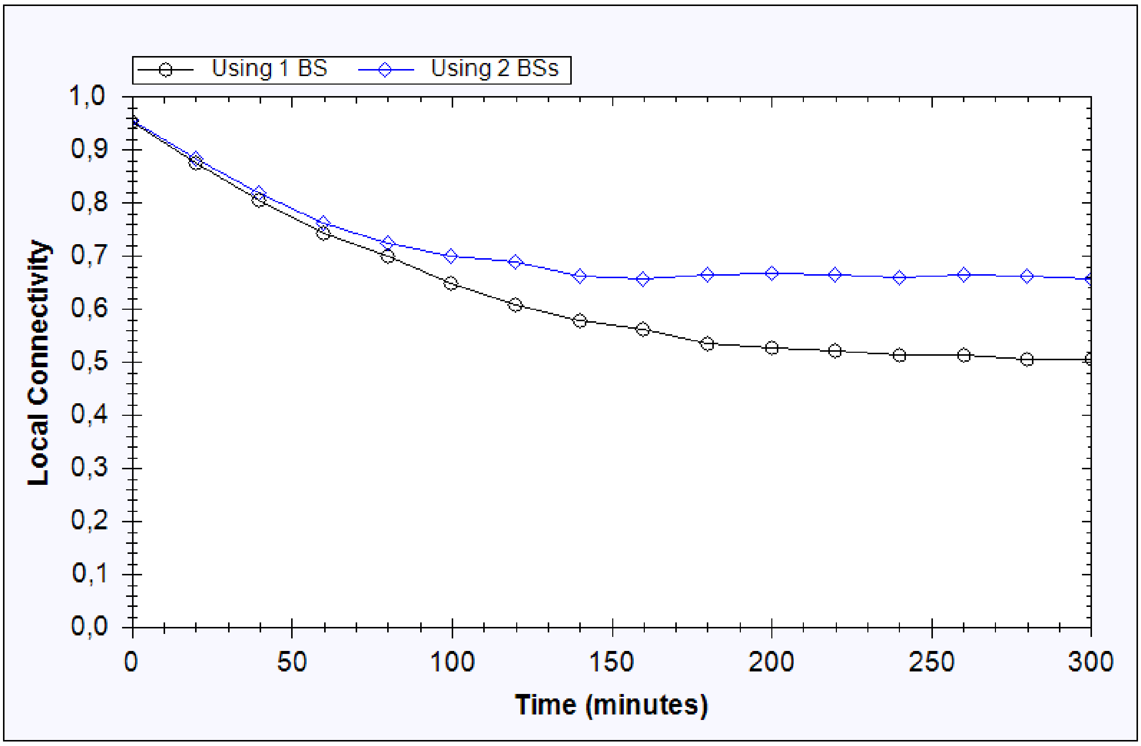


Figure 4.20 Local connectivity versus time using multiple BSs using Our Scheme + Du's Scheme for fully-mobile case where *BS speed*=400 meters/minute

4.2.2 Global Connectivity Performance

We also run simulations to calculate global connectivity. We use cases like using different *m* values, using different BS speeds and using multiple BSs. The results of the simulations are explained in the following subsections.

4.2.2.1 Global Connectivity for Different *m* Values

We use different *m* values of 100, 150, 200 and 250 and calculate global connectivity for each simulation over time. Since we use Du's Scheme at first, we could achieve very high global connectivity values from the very beginning of the deployment. The global connectivity values for these cases for *time*=0 and *time*=300 are shown in Table 4.4.

Table 4.4 Global Connectivity values for half-mobile and fully mobile-cases for different *m* values using Our Scheme + Du's Scheme

Key Chain Size, *m*	**Time**	**Half-Mobile**	**Fully-Mobile**
m=100	0	0.9976	0.9987
	300	0.9917	0.9955
m=150	0	0.9996	0.9999
	300	0.9962	0.9988
m=200	0	1	1
	300	0.9969	0.9992
m=250	0	1	1
	300	0.9987	0.9996

Table 4.4 shows that even at the beginning of the deployment the network has very high connectivity. The start values are around Du's Scheme global connectivity values and the end values are around our scheme's global connectivity values. These is a slight decrease over time, because of the mobile nature of the network, some nodes can get isolated from the rest of the nodes. Like the previous schemes, in this scheme as well fully-mobile cases generally achieve a slightly better global connectivity compared to half-mobile cases.

4.2.2.2 Global Connectivity for Different BS Speeds

We calculated global connectivity values over time for different BS speeds. It can be seen in Table 4.5 that as BS speed gets higher, global connectivity value also slightly increases and fully-mobile case has a better connectivity than half-mobile case. The decrease in global connectivity over time is due to the mobile nature of the network which can cause some nodes to get isolated from the rest.

Table 4.5 Global connectivity values for half-mobile and fully mobile cases for different BS speeds using Our Scheme + Du's Scheme

BS speed	Time	Half-mobile	Fully Mobile
BS speed=200 meters/minutes	0	1	1
	300	0.9947	0.9972
BS speed=400 meters/minutes	0	1	1
	300	0.9969	0.9992
BS speed=600 meters/minutes	0	1	1
	300	0.9977	0.9995

4.2.2.3 Global Connectivity Using Multiple Base Stations

We calculated global connectivity values for multiple BS case as well. The global connectivity values at *time*=0 and *time*=300 for both half-mobile and mobile case are seen in Table 4.6. It can be seen that using multiple BS brings a little increase in the global connectivity value and fully mobile case performs better than half-mobile case in terms of global connectivity. The slight decrease in global connectivity over time is due to the mobile nature of the network which can cause some nodes to get isolated from the rest.

Table 4.6 Global connectivity values for half-mobile and fully mobile case for different BS speeds using Our Scheme + Du’s Scheme

BS number	Time	Half-mobile	Fully Mobile
Using 1 BS	0	1	1
	300	0.9988	0.9987
Using 2 BS	0	1	1
	300	0.9993	0.9999

4.2.3 Resilience

In Du’s Scheme there is a possibility that different pairs of nodes can have the same key. If one node from one of the pairs gets captures by the attacker, he/she can get access to the keys of that node and use the keys to compromise any other communication links that use those keys. Therefore, just like Basic Scheme, Du’s Scheme is also vulnerable in terms of resiliency. Since we incorporate Du’s Scheme into our own scheme, this property affects our scheme as well. To see how this affects our proposed scheme, we conduct simulations to calculate additional links ratio and total compromised links ratio. We have two attack models like we had for the previous scheme. In the first model the attacker captures a certain amount of keys all at once. This model simulates the worst-case scenario. In the second model, an attacker captures nodes one by one over time. This model simulates a more likely to occur attack.

4.2.3.1 Worst Case Attack Scenario

We fix m=200 for this case as well. Captured node count is 200 for the simulations and all the nodes are captured at once at the beginning of the simulations. Figure 4.21 and 4.22 show the results for half-mobile and mobile case for additional compromised links ratio and total compromised links ratio.

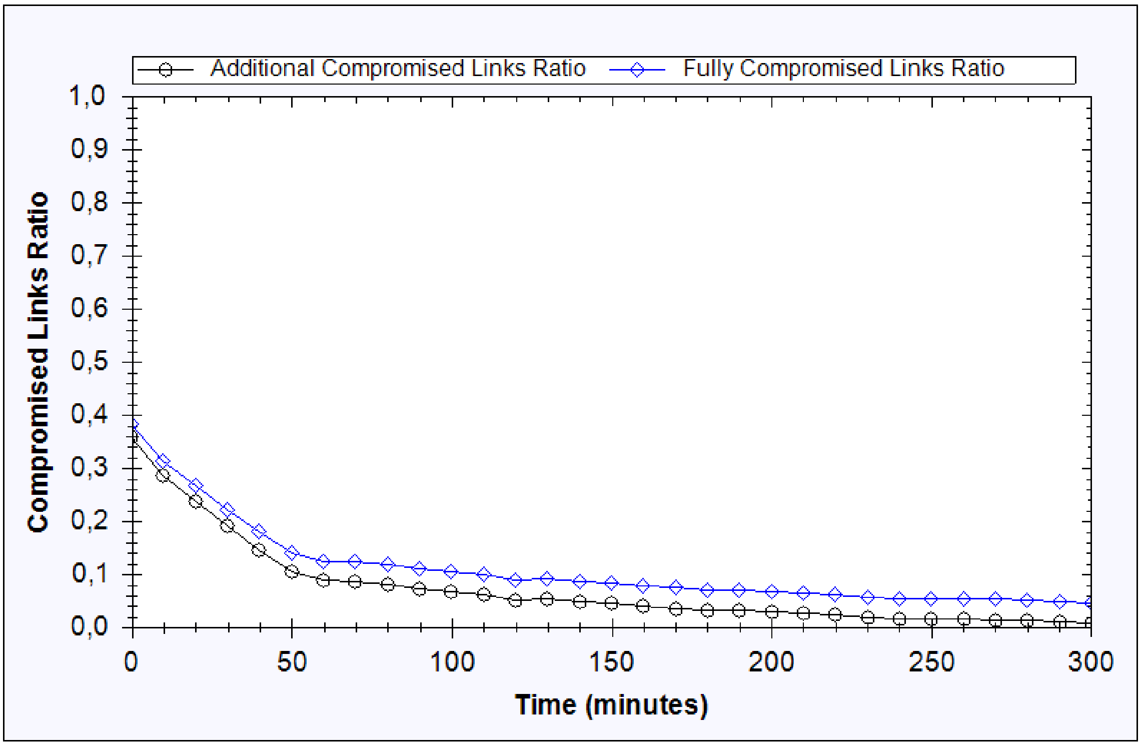


Figure 4.21 Additional and total compromised links ratio for captured node count=200 for half-mobile case using Our Scheme + Du's Scheme for worst case attack scenario

It can be seen from Figure 4.21 that additional compromised links ratio starts from Du's Scheme's value for this key chain size. Over time, both additional and total compromised links ratio decrease, since keys are being replaced by pairwise keys of our scheme. Towards the end of the simulations additional compromised links ratio gets closer to 0 and total compromised links get close to 0.04 which is the communication link ratio of total compromised links to all communication links in the network. Similar results can be seen for fully-mobile network. The results are shown in Figure 4.22. Please note that there is no significant difference between half-mobile and fully-mobile cases, because the improvement in resilience depends on the movement and speed of BS, not the mobility of the nodes.

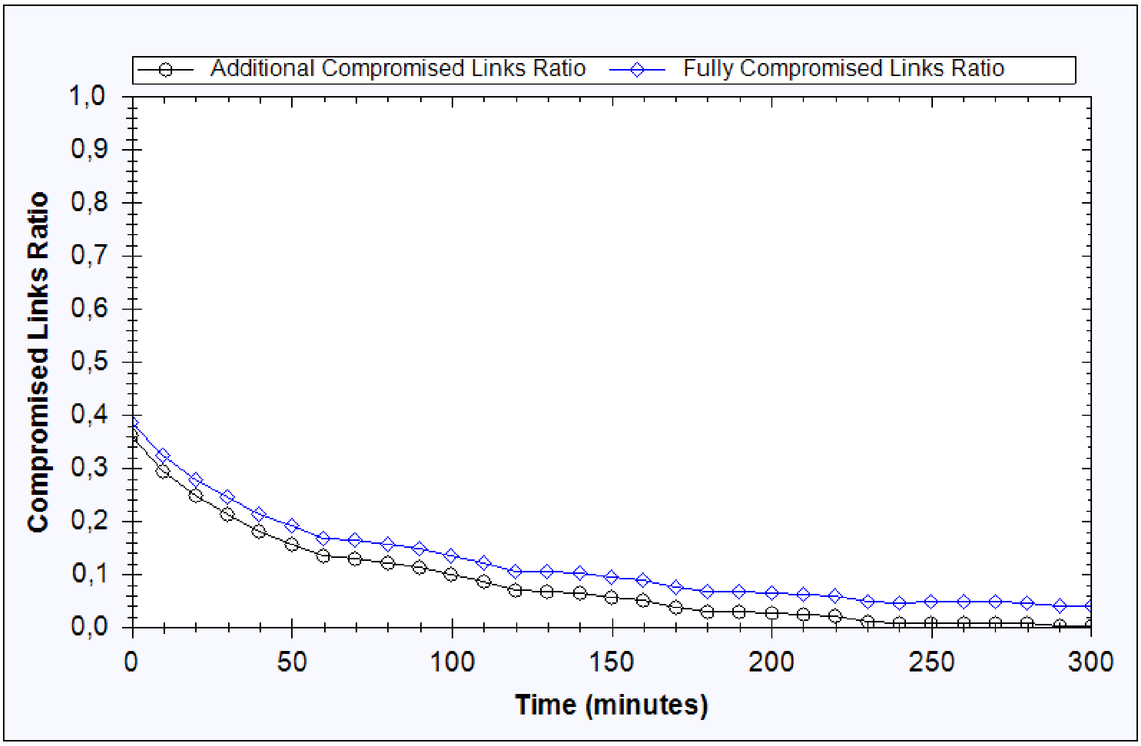


Figure 4.22 Additional and total compromised links ratio for captured node count=200 for fully-mobile case using Our Scheme + Du's Scheme for worst case attack scenario

4.2.3.1 Typical Attack Scenario

In this attack model, attacker captures the nodes gradually over time like the previous scheme. In this scenario we let the attacker capture one node per minute and the total amount of captured nodes are set to be 200. We kept *m* constant at 200 and BS speed constant at 400 meters/minute. Figure 4.23 and 4.24 show the compromised links ratio for half-mobile and fully-mobile cases.

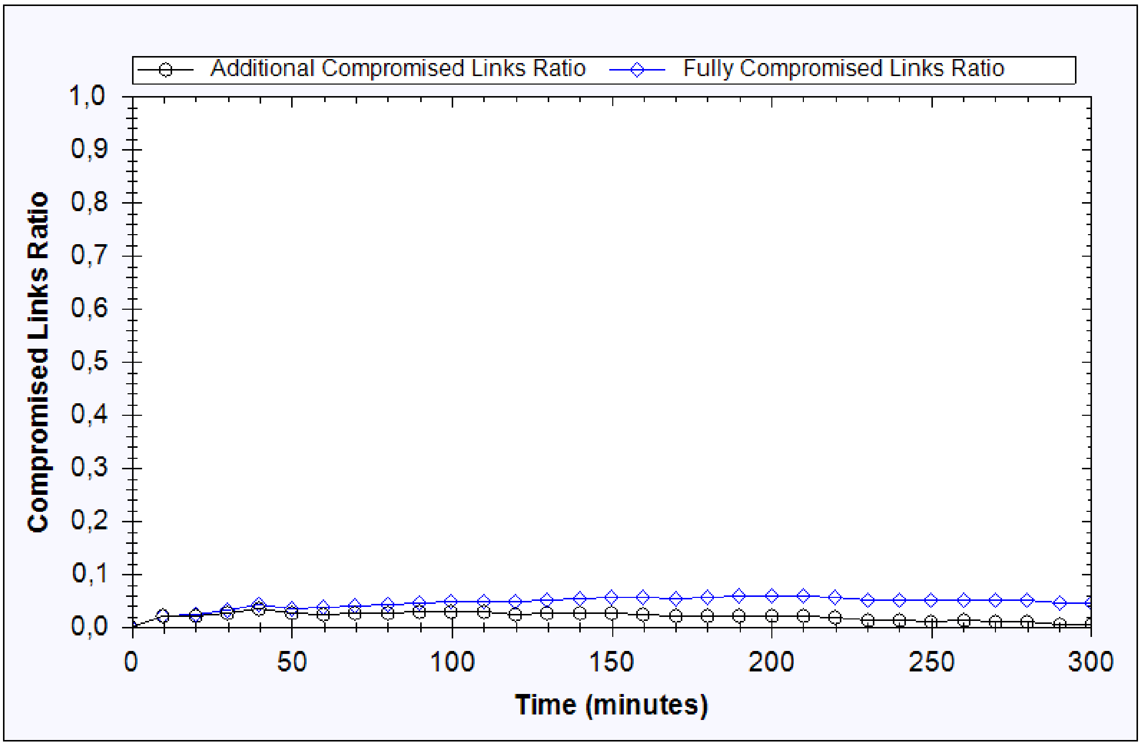


Figure 4.23 Additional and total compromised links ratio for captured node count=200 for half-mobile case using Our Scheme + Du’s Scheme for typical attac scenario

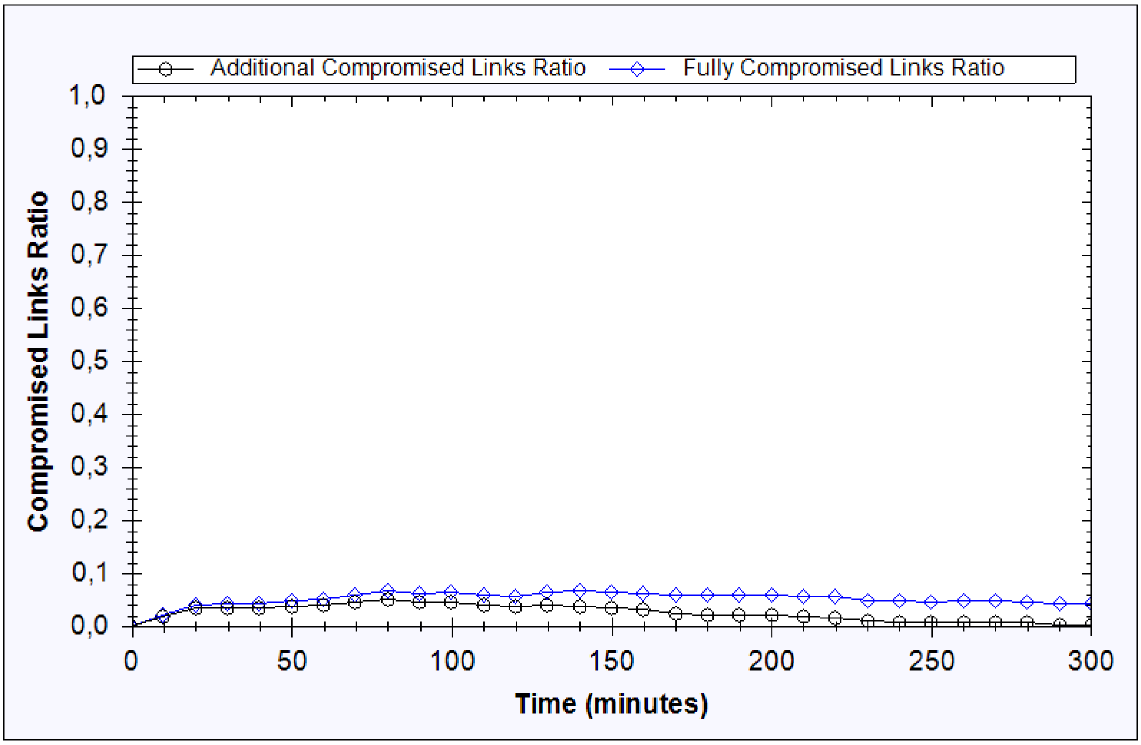


Figure 4.24 Additional and total compromised links ratio for captured node count=200 for fully-mobile case using Our Scheme + Du’s Scheme for typical attack scenario

It can be seen in Figure 4.23 and 4.24 that resilience of the scheme for this attack model is very good. Additional compromised links ratio increases a little at the beginning of the simulation; however it decreases and becomes close to 0 as time passes. This is because as time passes, the keys of Du’s Scheme are being deleted from the key chain. So even if an attacker captures nodes, it has very little help to him/her to compromising any additional links. Also note that total compromised links ratio converges to 0.04 which is the ratio of the total compromised links to all the links in the network.

Chapter 5

Conclusions

In this thesis, we propose a key distribution scheme for Mobile Wireless Sensor Networks and incorporate it with two existing schemes. We first look at existing key distribution schemes for Wireless Sensor Networks (WSNs) from mobility perspective. Our analysis show that most of the existing key distribution schemes do not take mobile WSNs in to account and propose solutions that work for static WSNs.

In Chapter 3, we first show the effects of mobility on two of the well known key distribution schemes, Basic Scheme by Esheneauer and Gligor and Du's Scheme by Du et al. We show that Du's Scheme, which uses deployment knowledge to achieve high connectivity is highly affected by mobility, where as Basic Scheme is not affected by mobility. After that we introduce our scheme which uses a mobile Base Station (BS). This mobile BS also works as a key distribution center. We run simulations for various scenarios like different key chain sizes, different BS speeds and using multiple BSs in the environment. Our simulations show that our scheme achieves a local connectivity value higher than the Basic Scheme for the corresponding key chain sizes. On the other hand, our scheme achieves a lower local connectivity value for the original Du's Scheme, but a higher value than the Du's Scheme shows when the nodes are mobile. It takes a while to achieve a convergence value in terms of local connectivity since nodes need to meet BS to get their keys. Local connectivity values for our scheme depends on metrics like, size of the key chain for each node, degree of the mobility of the nodes in the network, the speed of the BS and the number of BSs operating in the environment. For global connectivity, our scheme achieves values close to 1 as BS covers the whole simulation area. For resilience our scheme has perfect resilience against node capture since no additional communication link can be compromised even if the attacker captures some of the nodes in the environment.

In Chapter 4, we introduce two modifications to our original scheme to achieve high connectivity even at the beginning of deployment. In the first scheme, we use Basic Scheme at first and distribute nodes some keys according to Basic Scheme. After they are deployed and they meet with BS they slowly replace their key chains with the keys BS provides them. In the second scheme, we use Du's Scheme in a similar way. We start with Du's Scheme and slowly shift to our original scheme over time. Local connectivity values for these schemes start with respective schemes' original values and slowly converge to our scheme's values from Chapter 3. For global connectivity values the networks achieve a very high global connectivity even at the beginning so the nodes do not need to wait for BS to connect them and can start communication right from the beginning. Since both Basic Scheme and Du's Scheme can use same key for different pairs of nodes, they are vulnerable against node capture in terms of resilience. As we use these schemes, our scheme also becomes vulnerable. The ratio of the compromised links starts with the original values of respective schemes and over time they decrease and get closer to 0. As the keys from these schemes get eliminated resilience is improved as well.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002

[2] L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks. In *Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS'02), ACM,* New York, NY, USA, 2002, pp. 41–47.

[3] W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04),* vol. 1, IEEE Computer Society, Los Alamitos, CA, USA, 2004, pp. 586–597.

[4] R. Blom, An optimal class of symmetric key generation systems, in: *Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques,* Springer, New York, NY, USA, 1985, pp. 335–338.

[5] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, A. Khalili, A pairwise key pre-distribution scheme for wireless sensor networks, in: *Proceedings of the 10th ACM conference on Computer and communications security (CCS'03),* ACM, New York, NY, USA, 2003, pp. 42–51

[6] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly secure key distribution for dynamic conferences, in: *LNCS*, vol. 740, Springer, New York, NY, USA, 1993, pp. 471–486.

[7] S. Çamtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, in: *Proceedings of the Ninth European Symposium on Research Computer Security,* IEEE Press, Piscataway, NJ, USA, 2004, pp. 293–308.

[8] S. A. Munir, R. Biao, J. Weiwei, W. Bin, X. Dongliang, M. Man, Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol.2, no., 21-23 May 2007, pp.113-120.

[9] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley, Mobility improves coverage of sensor networks. In *Proceedings of ACM MobiHoc 2005.*

[10] G. Wang, G. Cao, T. L. Porta, Movement-assisted sensor deployment. In *Proceedings of IEEE* Infocom, 2004.

[11] A. Howard, A. Mataric, A. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS 02).* Fukuoka, Japan June 25-27, 2002.

[12] J.P. Walters, Z. Liang, W. Shi, V. Chaudhary, Wireless sensor network security: a survey. In Security in Distributed, Grid, and Pervasive Computing, Auerbach Publications, CRC Press

[13] Y. Zhou, Y. Fang, Y. Zhang, Securing wireless sensor networks: a survey. *Communications Surveys & Tutorials, IEEE,* vol.10, no.3, pp.6-28, Third Quarter 2008.

[14] G. Gaubatz, J.-P. Kaps, E. Öztürk, B. Sunar, State of art in ultra low-power public key cryptography for wireless sensor networks. In *PerCom Workshops,* 2005, pp. 146-150.

[15] F. Amin, A. H. Jahangir, H. Rasifard, Analysis of public key cryptography for wireless sensor netwoks security. In *PWASET '08: Proceedings of World Academy of Science, Engineering and Technology*, July 31, 2008.

[16] B. Arazi, I. Elhanany, O. Arazi, H. Qi Revisiting public-key cryptography for wireless sensor networks. *Computer,* col. 38, no 11, pp. 103-105, 2005.

[17] S. A Çamtepe, B. Yener, *Key Distribution Mechanisms for Wireless Sensor Networks: a Survey.* Technical Report TR-05-07 Rensselaer Polytechnic Institute, Computer Science Department, March 2005. Book Chapter: Key Management in the book Wireless Sensor Networks Security, IOS Press.

[18] J.Zhang, V. Varadharajan, Wireless sensor network key management survey and taxonomy. *Journal of Network and Computer Applications*, 2009.

[19] J. C. Lee, V. C. M. Leung, K. H. Wong, J. Cao, H. C. B Chan, Key management issues in wireless sensor networks: current proposals and future developments. *IEEE Wireless Communications*, vol. 14, no. 5, pp. 76-84., 2007.

[20] Y. Xiao, V. K. Rayi, B.Sun, X.Du, F.Hu, M. Galloway, A survey of key management schemes in wireless sensor networks. *Comput. Commun.* 30, 11-12 (Sep. 2007), 2314-2341.

[21] M. A. Simplício, Jr., M. Barreto, B. C. Margi, T. Carvalho. 2010. A survey on key management mechanisms for distributed Wireless Sensor Networks. *Comput. Netw.* 54, 15 (October 2010), 2591-2612.

[22] B. Lai, S.Kim, I. Verbauwhede, Scalable session key construction protocol for wireless sensor networks. In: *IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES),* IEEE Computer Society, Washington, DC, USA, 2002.

[23] ZigBee Alliance, Zigbee specification document 053474r06, v1.0. Technical report, ZigBee Alliance, 2004.

[24] Y. Zeng, B. Zhao, J. Su, X. Yan, Z. Shao, A loop-based key management scheme for wireless sensor networks. In: *Emerging Directions in Embedded and*

Ubiquitous Computing (EUC Workshops), LNCS, vol. 4809, Springer, Berlin/Heidelberg, 2007, pp. 103–114.

[25] H. Chan, A. Perrig, D. Song, Random key pre-distribution schemes for sensor networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP'03),* IEEE Computer Society, Washington, DC, USA, 2003, pp. 197–213.

[26] S. Hussain, M. Rahman, L. Yang, Key pre-distribution scheme using keyed-hash chain and multipath key reinforcement for wireless sensor networks, IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 1–6.

[27] T. Shan, C. Liu, Enhancing the key pre-distribution scheme on wireless sensor networks, in: *IEEE Asia-Pacific Conference on Services Computing*, IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 1127–1131.

[28] C.-F. Law, K.-S. Hung, Y.-K. Kwok, A novel key redistribution scheme for wireless sensor networks, in: *IEEE International Conference on Communications (ICC'07),* IEEE Computer Society, Washington, DC, USA, 2007, pp. 3437–3442.

[29] S. Zhu, S. Xu, S. Setia, S. Jajodia, Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach, in: *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03),* IEEE Computer Society, Washington, DC, USA, 2003, pp. 326–335.

[30] C. Castelluccia, A. Spognardi, RoK: a robust key pre-distribution protocol for multi-phase wireless sensor networks, in: *Proceedings of the Third International Conference on Security and Privacy in Communications Networks (SecureComm'07),* IEEE Computer Society, Los Alamitos, CA, USA, 2007, pp. 351–360.

[31] M. Ergun, A. Levi, E. Savas, A resilient key pre-distribution scheme for multiphase wireless sensor networks, in: *Proceedings of the 24th International Symposium on Computer and Information Sciences (ISCIS'09),* IEEE Computer Society Washington, DC, USA, 2009, pp. 375–380.

[32] J. Lee, D. Stinson, Deterministic key pre-distribution schemes for distributed sensor networks, in: *LNCS – SAC'2004*, vol. 3357, Springer Berlin/Heidelberg, 2005, pp. 294–307.

[33] H. Chien, R.-C. Chen, A. Shen, Efficient key pre-distribution for sensor nodes with strong connectivity and low storage space, in: *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA'08),* IEEE Computer Society, Washington, DC, USA, 2008, pp. 327–333.

[34] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: *Proceedings of the 10th ACM Conference on Computer and communications Security (CCS'03),* ACM, New York, NY, USA, 2003, pp. 52–61.

[35] A. Gupta, J. Kuri, Deterministic schemes for key distribution in wireless sensor networks, in: *Proceedings of the Third International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08),* IEEE Computer Society, Washington, DC, USA, 2008, pp. 452–459.

[36] D. Liu, P. Ning, Location-based pairwise key establishments for static sensor networks, in: *Proceedings of the First ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03),* ACM, New York, NY, USA, 2003, pp. 72–82.

[37] D. Liu, P. Ning, Improving key pre-distribution with deployment knowledge in static sensor networks, *ACM Transactions on Sensors and Networks 1* (2) (2005) 204–239.

[38] K. Kalkan, S. Yilmaz, O. Yilmaz, A. Levi, A highly resilient and zone based key pre-distribution protocol for multiphase wireless sensor networks, in: *Proceedings of the Fifth ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet'09),* ACM, New York, NY, USA, 2009, pp. 29–36.

[39] Z. Yu, Y. Guan, A key management scheme using deployment knowledge for wireless sensor networks, IEEE *Transactions on Parallel Distribution and Systems 19* (10) (2008) 1411–1425.

[40] D. Huang, M. Mehta, D. Medhi, Harn, L. Location-aware key management scheme for wireless sensor networks. In *2nd ACM workshop on Security of Ad Hoc and Sensor Networks,* 2004.

[41] N. Canh, P. Truc, T. Hai, L. Hung, Y. Lee, S. Lee, Enhanced group-based key management scheme for wireless sensor networks using deployment knowledge, in: *Proceedings of the Sixth IEEE Consumer Communications and Networking Conference (CCNC'09),* IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 1–5.

[42] S. Čapkun, J. P. Hubaux, L. Buttyán, Mobility helps security in ad hoc networks. In *Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking &Amp; Computing* (Annapolis, Maryland, USA, June 01 - 03, 2003). MobiHoc '03. ACM, New York, NY, 46-56.

[43] S. Čapkun, J. P. Hubaux, L. Buttyán, Mobility helps peer-to-peer security. *Mobile Computing, IEEE Transactions on* , vol.5, no.1, pp. 43- 51, Jan. 2006

[44] L. Zhou, J. Ni, C.V. Ravishankar, Efficient key establishment for group-based wireless sensor deployments. In *Proceedings of the 4th ACM Workshop on Wireless Security* (Cologne, Germany, September 02 - 02, 2005). WiSe '05. ACM, New York, NY, 1-10.

[45] B. Zhou, K. Xu, K. Gerla, Group and swarm mobility models for AD HOC network scenarios using virtual tracks. In *Military Communications Conference (MILCOM),* 2004.

[46] A. Ünlü, A. Levi, Two-tier, scalable and highly resilient key predistribution scheme for location-aware wireless sensor network deployments. *Mob. Netw. Appl.* 15, 4 (Aug. 2010), 517-529.

[47] W. Du, J. Deng, Y. S. Han, P. Varshney, A pair-wise key predistribution scheme for wireless sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*. pp 42–51

[48] Q. Dong, D. Liu, Using auxiliary sensors for pair-wise key establishment in WSN. In *Proceedings of the 6th international Ifip-Tc6 Conference on Ad Hoc and*

Sensor Networks, Wireless Networks, Next Generation internet (Atlanta, GA, USA, May 14 - 18, 2007). I. F. Akyildiz, R. Sivakumar, E. Ekici, C. J. De Oliveira, and J. McNair, Eds. *Lecture Notes In Computer Science. Springer-Verlag,* Berlin, Heidelberg, 251-262.

[49] R.M. Needham, M.D. Schroeder, Using encryption for authentication in large networks of computers. *Commun. ACM 21(12)* 993–999. 1978

[50] T. Camp, J., Boleng, V., Davies, A survey of mobility models for AD hoc network research. In *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483-502, 2002.

[51] C. Chen, J. Ma, Mobile Enabled Large Scale Wireless Sensor Networks. *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference* , vol.1, no., pp.333-338, 20-22 Feb. 2006

[52] C. Chen, J. Ma, MEMOSEN: multi-radio enabled mobile wireless sensor network. *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on* , vol.2, no., pp. 5 pp., 18-20 April 2006.