Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5531923, 20 pages
https://doi.org/10.1155/2021/5531923

WILEY | Hindawi

Research Article

A Spontaneous Wireless Ad Hoc Trusted Neighbor Network Creation Protocol

Jose Vicente Sorribes, Lourdes Peñalver, and Jaime Lloret

Universitat Politecnica de Valencia, Valencia, Spain

Correspondence should be addressed to Jaime Lloret; jlloret@dcom.upv.es

Received 5 February 2021; Revised 3 May 2021; Accepted 24 May 2021; Published 7 July 2021

Academic Editor: Carlo Giannelli

Spontaneous networks lack an a priori communication infrastructure, the neighbors are unknown right after the deployment, and they are used during a period of time and in a certain location. In this paper, we present a new randomized creation model of a spontaneous wireless ad hoc network based on trusted neighbors. The idea is to manage the neighbor discovery with the exchange of identity cards, and the checking of a signature establishes a relationship based on trust of the neighbors. To asses the performance of our randomized trusted network proposal and compare it against an existing deterministic protocol used as reference, we relied on Castalia 3.2 simulator, regarding 4 metrics: time, energy consumption, throughput, and number of discoveries vs packet sent ratio. We found that our proposal outperforms the reference protocol in terms of time, energy, and discoveries vs packet sent ratio in a one-hop setting, while it outperforms the reference protocol regarding all 4 metrics in multihop environments. We also evaluated our proposal through simulations varying the transmission probability and proved that it does not require to know the number of nodes if a fixed transmission probability is set, providing reasonable results. Moreover, our proposal is based on collision detection, it knows when to terminate the process, it does not require a transmission schedule, and it follows more realistic assumptions. In addition, a qualitative comparison is carried out, comparing our proposal against existing protocols from the literature.

1. Introduction

In spontaneous ad hoc networks, a concept introduced in [1], the devices (also known as nodes) that conform them are autonomous and equipped with a limited transmission range radio transceiver. Therefore, some nodes can communicate directly with nodes in their transmission range (the neighbors). However, other nodes need multiple intermediate nodes that forward the information that is not addressed for their own use in a multihop fashion. For this purpose, each node must act as a router [2, 3].

This type of networks does not have a communication infrastructure right after the deployment. Each node does not know its neighbors. Thus, in the creation of a spontaneous ad hoc network, the neighbor discovery [4, 5] becomes necessary to find out which nodes are within transmission range. Furthermore, there is absence of a central server; so there is no centralized CA (Certificate Authority) available and each node must act as a CA.

Spontaneous networks are a special kind of ad hoc networks which present some characteristics [6], such as the new services will be available without user intervention; the nodes can meet in a physical location in a certain amount of time, collaborating in every moment to provide services such as group communication, security, and so on; the nodes can join or leave the network at will at any time and the devices can come from everywhere; these networks are conformed by mobile nodes; thus, there is no fixed topology; they emulate the human relations to manage the creation and operation; they are conformed by a set of nodes that sometimes do not know each other; these networks must have a security level similar to the traditional wired networks; each node acts as a router, the nodes have a limited communication range towards other nodes; they have limited resources

including CPU, memory, and energy (batteries); mobile nodes can move freely in the given area even out of each other's range; the physical transmission medium is shared; the different identities are given by IP addresses dynamically obtained; and there is no central administration.

To summarize, a spontaneous ad hoc network is different from an ad hoc network since it is used in a certain location during a period of time, which does not depend on a central server, and the user is not required to be an expert, imitating human relationships in order to work together in groups, with minimal user intervention.

Usually, this type of networks uses trust relationship [7], imitating how humans interact, in the creation and management, building a trust chain (also known as "trust net").

In addition, later on, when the spontaneous network is already created and ad hoc routing is necessary, if a node trusts a second node, it can send messages directly to it. In case that the second node does not trust the first node, the communication is not allowed. When a node wants to send a message towards a nontrusted node, it has to do it through a trusted node.

In this context, we would like to build a spontaneous ad hoc network based on trust. The idea is to use human relationships as a model, following a scenario that takes place for instance when a group of humans join to communicate, exchange information, or work together for a period of time in a certain location.

Many protocols from the literature (i.e., the deterministic approaches) need a transmission schedule, while some randomized protocols require unrealistic assumptions such as not using the collision detection and ignore the termination condition. Thus, the main objective of our work is to propose and evaluate protocols which do not rely on a transmission schedule, deal with collisions, operate under more realistic assumptions, and obtain better performance results than existing solutions.

The main goal of creating a spontaneous ad hoc network is to establish a distributed key management service through the use of a network of trust. Therefore, public keys will only need to be obtained when necessary in further operations.

There are many application areas [8] for spontaneous ad hoc networks which include industrial (e.g., communication among sensors, robots, and digital networks), business (e.g., meeting, stock control), military (e.g., hard and hostile environments), and teaching. Many possible examples can be mentioned for this type of networks, such as wireless sensors in a forest to detect fire for a certain period of time, sensors in a bridge aiming at counting the number of vehicles and their speed, or sensors deployed in a lake in order to study the water quality in a certain period of time.

In this paper, a protocol for the creation of static spontaneous wireless ad hoc networks based on trust is presented and implemented in Castalia 3.2 simulator [9] for validation and comparison purposes. The target networks are static; i.e., the nodes cannot move in the deployment area. Therefore, the proposal could be improved to be used in dynamic networks if it properly takes into account node joining and leaving the network and nodes getting in and out of each other's transmission range. The proposal combines neighbor discovery with identity card exchange and signature checking to establish a network based on trust. The identity card exchange allows to disseminate the public keys throughout the network. After this exchange, the signature is checked using the public key and if it is okay, the neighbor is considered as trusted. This type of neighbors create thus a network of trust. Our proposal is compared against an existing solution. We relied on the Castalia 3.2 simulator for validation and comparison purposes and used four metrics: time, energy, throughput, and number of discoveries vs packet sent ratio.

The problem that our proposal is aimed at solving is that the solution has to cope with channel collisions and still discover all the neighbors, succeed at exchanging identity cards, and discover all the trusted neighbors, without relying in a transmission schedule, improving the time, energy consumption, throughput, and number of discoveries vs packets sent compared to previous works, and other solutions need to know the number of nodes in the network. Other randomized related works could not address it since the protocols do not take into account the collisions or do not handle them and do not know when to terminate. We introduce a probabilistic mechanism that copes with collisions and discovers all the neighbors.

The problem statement we must cope with by introducing a proposal is that the nodes will operate in static environments, the devices are equipped with limited range radio transceivers, the devices use half-duplex mode only, the nodes are randomly deployed in a given area, the nodes should be asynchronous and be aware of channel collisions, the nodes can detect them, the number of nodes in the network must be unknown, the nodes must discover all their neighbors with high probability (nearly 1) and know when to terminate the protocol, and the solution must not rely on a transmission schedule and obtain better performance results than existing solutions.

At the end of this work, we will list the differences of our proposal and existing works in the literature in Table 1.

The novelty of this work compared to prior works is that a schedule is not used, a priori knowledge of the number of nodes in the network is not required, and the protocol is tailored for static environments.

The main contributions of this work are as follows: (i) a probabilistic two-phase proposal that manages to cope with and detect collisions, allows termination detection, can use a fixed transmission probability (i.e., it does not require to know the number of nodes), does not depend on a transmission schedule, discovers all the neighbors, succeeds at exchanging the identity cards, discover all the trusted neighbors with probability almost 1, and follows more realistic assumptions, and it is suitable to be used both in one-hop and multihop environments; (ii) a qualitative comparison of related work protocols and our proposal is available; (iii) an implementation in Castalia 3.2 and comparison of our proposal against a reference protocol are also available; and (iv) a study of the behavior of our proposal varying the transmission probability. Furthermore, we found that the proposal is faster and spends less energy than existing solutions.

According to the simulation results, we found that our proposal outperforms the reference protocol in terms of time,

Table 1: Qualitative comparison of related work protocols and our proposal.

	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[13]	[23]	Proposal
IoTs	Yes	No	No	No	No	No	No	No	No	No
Cloud network	No	Yes	No	No	No	No	No	No	No	No
Call to network goes to a web server connected to IP cloud	Yes	No	No	No	No	No	No	No	No	No
Mobile network	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Spontaneous wireless ad hoc networks	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Manages network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Create resources	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Share secure data	No	No	No	Yes	Yes	No	Yes	Yes	Yes	No
Share services and resources	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Offer secure services	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Developed prototype	Yes	Yes	Java (J2ME) with KVM	No	No	No	No	No	No	No
Real deployment	No	No	Mobile Nokia E65	No	No	No	No	No	No	No
Devices with limited resources	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Heterogeneous systems (different devices)	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Communities with low resources	Yes	No	No	No	No	No	No	No	No	No
Device with unique identity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Community with unique group identity	Yes	No	No	No	No	No	No	No	No	No
Simulation	Castalia/OPNET	Castalia	Yes	No	No	No	No	No	No	Castalia
Neighbor discovery phase	Yes	Yes	No	No	No	No	No	No	No	No
Neighbor threshold	Yes	Yes	No	No	No	No	No	No	No	No
Neighbor card list	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Identity card	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Public-private key pair	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Certificate signed by private key	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Identity card exchange	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Summary function hash	Yes	Sha-1	Sha-1	Yes	Yes	SHA-1	SHA-1	SHA-1	SHA-1	No
Local repository of public key certificates and trust values	No	Yes	Yes	No	No	Yes	No	No	No	Yes
Minimal user interaction (user-friendly application)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Users not experts	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Authentication phase	No	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Preauthentication phase	Yes	No	No	No	No	No	No	No	No	No
Trust established by a user	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Trust established automatically	Yes	No	No	No	No	No	Yes	Yes	Yes	Yes
Preauthentication user decides trust level	Yes	No	No	No	No	No	No	No	No	No
Trust chain	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ranges of trust	Yes	No	No	No	No	No	No	No	No	No
Only two trust levels	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Changing trust values	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No

Table 1: Continued.

	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[13]	[23]	Proposal
Trust modifiable based on behavior	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No
Can also stop trusting	No	No	No	No	No	Yes	Yes	Yes	No	No
Public keys obtained through network of trust	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Distributed administration service through trusted network	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Public key infrastructure	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Each node acts as client/server	Yes	Yes	Yes	No	No	Yes	No	Yes	No	No
Nodes join and leave network at will at any time	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
First node creates and sets up the network	No	No	Yes	No	Yes	No	Yes	Yes	Yes	No
Internet access to all nodes	Yes	Yes	No	No	Yes	Yes	No	No	No	No
Share many Internet services	No	Yes	No	No	Yes	No	No	No	No	No
Collaborate within Intranet/Internet	Yes	Yes	No	No	Yes	Yes	No	No	No	No
Distributed networks	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Central administration	No	No	No	No	No	No	No	No	No	No
Independent networks	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Self-configured network	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Trust value based on human relations	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Devices similar behavior to human relationships	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Predistribution key algorithm	No	No	No	Yes	No	No	No	No	No	No
Symmetric key encryption	No	AES	AES	AES	Yes	AES	Yes	AES	AES	No
Asymmetric key encryption	No	RSA/ECC	RSA/ECC	Yes	Yes	RSA/ECC	Yes	RSA	RSA/ECC	No
Cryptography suitable for low-power devices	No	No	No	No	No	Yes	Yes	Yes	Yes	No
Session key	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
Distributed CA (each node CA)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Distributed name service	No	No	No	No	No	Yes	No	No	No	No
The CA for a node any of the trusted nodes	YES	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Trust relationship can be asymmetric	No	No	No	No	No	Yes	Yes	Yes	No	No
Trust based on physical proximity (first visual contact)	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Unique IP address assignment	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Detect existence of duplicated IP addresses	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	No
Addresses obtained dynamically	No	Yes	No	No	No	No	Yes	Yes	Yes	No
Authenticate using IP addresses	No	No	Yes	No	No	No	No	Yes	Yes	No
Authentication exchanging keys	No	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Authorization exchanging keys	No	No	No	Yes	No	Yes	Yes	Yes	Yes	No
Key management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Initial exchange of data/keys	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Exchange secret keys to encrypt data	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 1: Continued.

	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[13]	[23]	Proposal
Each node requests services from its trusted nodes	No	No	No	Yes	No	Yes	Yes	No	Yes	No
Groups work in a collaborative way for the network maintenance	Yes	No	No	No	No	Yes	Yes	Yes	Yes	No
Just one node is required to be connected to the Internet	Yes	Yes	No	No	Yes	Yes	No	No	No	No
More than one node can be connected to provide Internet access	Yes	No	No	No	Yes	Yes	No	No	No	No
Connection shared if one user has Internet connection	Yes	Yes	No	No	Yes	Yes	No	No	No	No
Access to the WWW if one user has Internet connection	Yes	No	No	No	Yes	Yes	No	No	No	No
Best nodes carry out communications through the Internet	Yes	No	No	No	Yes	Yes	No	No	No	No
Services shared using TCP connections	No	No	Yes	No	No	No	No	No	No	No
TCP/IP protocols	Yes	No	No	No	No	No	No	No	No	No
Network built using IEEE 802.11b/g	No	No	Yes	No	No	No	No	Yes	Yes	No
Authentication through Bluetooth or ZigBee	No	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Based on social networks	Yes	No	No	No	No	No	No	No	No	Yes
Intrusion detection technique	No	No	No	Yes	Yes	No	No	Yes	No	No
Caching technique to avoid overload of the nodes	No	No	No	No	Yes	No	No	No	No	No

energy consumption, and discoveries vs packet sent ratio in a one-hop setting, whereas it outperforms the reference protocol regarding all 4 metrics, i.e., time, energy consumption, throughput, and discoveries vs packet sent ratio, in multihop environments. We also focused on the evaluation of our proposal by varying the transmission probability and demonstrated that the proposal does not require to know the number of nodes when a fixed transmission probability is set, providing good results.

A list of key concepts and their definitions is provided at the end of this paper.

The rest of the paper is organized as follows: a brief related work can be found in Section 2. Our proposal, assumptions, and model are presented in Section 3. An overview of the reference protocol, the simulation setup, and the simulation results are shown and discussed in Section 4. A qualitative comparison of protocols and our proposal takes place in Section 5. Some concluding remarks are made in Section 6. Finally, an appendix which includes the definition of the main key concepts is available in the appendix.

2. Related Work

There are many works related to spontaneous networks in the literature. Next, we will describe some of them.

Authors explain in [1] the difference between ad hoc networks and spontaneous networks. They identify 5 key challenges introduced by the environment of spontaneous networks. One of the main points that make a difference between a spontaneous network and fixed or mobile networks is that they make easy the integration of services and devices, fixing new services and configuration parameters of devices. They must be carried out without the user intervention or interference in the network operation. The bad operation or failure in one of the devices or services does not compromise the feasibility of the community.

In [10], the authors proposed SCOPE: a prototype for spontaneous P2P social networks. Below the network layer, SCOPE follows the 802.11 ad hoc mode and it does not need infrastructure. SCOPE follows the hierarchical P2P model. Some nodes with higher calculation capacity become supernodes. The supernodes form an overlay and provide the distributed data management system for the P2P social network. The client nodes connect with supernodes and rely on them to share their contents or access the shared information.

A method to join spontaneous networks is proposed in [11]. The authors present a proposal to implicit join spontaneous networks following a group mobility mode. The routing protocol between cells avoids bottle necks in the links. Some hierarchical routing protocols are based on the election of a cluster-head cell (or reference point node).

Some examples about IoTs (Internet of Things) can be found in the HP Labs CeNSE project [12]. The authors focus on the deployment of a worldwide sensor network to create a

"central nervous system for the Earth." They also center on the "A Smarter Planet" project, a strategy developed by IBM that considers the sensors as fundamental basis in smart systems of water management and smart cities.

A complete secure protocol for spontaneous wireless ad hoc networks, based on the trust between collaborating nodes, is presented in [13]. It uses symmetric encryption AES, while asymmetric key scheme uses RSA, for distributed user authentication. It is designed to be used in mobile devices with limited resources and requires limited memory space and energy. It provides an intrusion detection mechanism, used to detect the nodes which can be attackers. An authenticated user can display the nodes, update the information, process an authentication request, reply to an information request, send data to one node, or leave the network.

In [14], the authors focus on a specific challenge: the current connectivity model between the WSN (wireless sensor network) and Internet. The authors try to answer if the sensor nodes should delegate all the communications of the Internet to a set of central management systems or if they should convert into first-class citizens of the Internet implementing the entire TCP/IP stack as well as other standards as web services.

A review on a secure protocol for spontaneous wireless ad hoc networks is presented in [15], focusing on devices with limited resources. According to the protocol, services and resources can be exchanged. Security for this type of networks uses hybrid symmetric/asymmetric key management scheme to exchange the data, identity cards are encrypted before the exchange, and symmetric key scheme is used to encrypt the data. The symmetric key cryptography scheme uses AES algorithm. Certificate exchange will be encrypted by using asymmetric key cryptography scheme ECC algorithm because of its better results. Trust is formed by visual contact or by an authentication procedure using a session key.

In [16], a secure spontaneous wireless ad hoc network creation protocol to access the IoTs, based on direct P2P interaction and communities, is presented, and it is used by different types of devices with limited resources. The protocol is aimed at improving the communication within the Intranet and on the Internet and integration among different low-resource communities. This protocol allows users to securely access to the WWW by shared Internet connection between communities through a single or several nodes that use TCP/IP. Simulations have been performed using the Castalia simulator. In the experiments, a web server connected to an IP cloud simulates Internet behavior and different spontaneous networks are also connected up to this IP cloud. According to the simulation results, a 61% improvement is obtained against a conventional architecture; the traffic is more stable and displays fewer fluctuations. Finally, a prototype is available in [16].

A spontaneous mobile ad hoc cloud computing network creation protocol is presented in [17], to share computing resources and applications. A distributed CA service is proposed. The security management is based on a public key infrastructure for user authentication, in which each user maintains a local repository of public key certificates and their trust values. The proposal uses a summary SHA-1; an asymmetric key encryption scheme uses RSA and ECC, mainly used in the user authentication process. Symmetric key encryption uses the AES algorithm and is used as a session key. Bluetooth is used in the authentication process. Simulation results have been obtained using the Castalia 2 simulator, achieving good efficiency and performance even with a high number of nodes. A prototype has been implemented to simulate the creation of a mobile cloud computing system using a spontaneous ad hoc network.

A secure self-configured protocol [18] for distributed and decentralized spontaneous wireless ad hoc network creation and management is presented and focuses on low-power devices. When a new node joins the network, it uses an identity card, hash SHA-1, and certificate. The AES algorithm has been chosen for symmetric encryption scheme. On the other hand, ECC and RSA have been used for asymmetric encryption scheme. Further, when the network has been created, services are shared by means of TCP connections using IEEE 802.11b/g technology. Bluetooth or ZigBee allows authentication of nodes when they join the network. A prototype has been developed using Java (J2ME) programming. A real deployment in a mobile device Nokia E65 performing a spontaneous network is available. Several tests have been carried out to validate the protocol operation and compare the protocol with other spontaneous ad hoc network protocols. The response times obtained are suitable for its use in real environments, even when devices have limited resources. Authors found that storage and volatile memory needs are quite low and the protocol can be used in regular resource-constrained devices.

A secure completely self-configured protocol [19] for the creation of spontaneous wireless ad hoc networks is presented. It uses a predistribution key based on user trust in order to exchange initial data and the secret keys; it shares services and resources and focuses on devices with limited resources. A user can create its own resources, or it can request them from its neighbors. To achieve node authentication, key exchange mechanism is required. In the network creation, the first step takes place when a new node joins the network and exchanges identity cards. Then, a service accessing phase takes place. Finally, a trust chain is formed. Furthermore, the proposal uses the AES algorithm for a symmetric encryption scheme and an asymmetric scheme. According to the simulation results, execution times and energy consumption can be improved by this protocol. Moreover, a remarkable advantage is that authors present an intrusion detection technique.

A complete self-configured secure symmetric key protocol [20] for independent and decentralized spontaneous mobile wireless ad hoc network creation and management is presented. It is aimed at improving communication and integration between different study centers of low-resource communities. This protocol is used to share resources and many Internet services to the whole network, where only one node is connected to the Internet. An intrusion detection scheme for joining members is used. This proposal uses asymmetric cryptography for device identification and symmetric cryptography to share session keys. In a joining step, the system handles the identity cards and certificates. Public

key infrastructure is carried out, and the public key is used as a session key. Devices must collaborate within the Intranet or on the Internet. The connection can be shared, and the first node in the network will provide access to the WWW if it has Internet connection. However, for the Internet access, there could be more than one node and each node can share different services. The authors in [20] show the design and simulation of a model that lets optimal spontaneous network access by using a caching mechanism. An analytical proposal, a validation through simulations and comparison with regular architectures and the most similar protocols from the literature, is available in [20].

A complete self-configured security based on user trust protocol [21] for distributed and decentralized spontaneous mobile wireless ad hoc network creation and management is presented. This protocol allows to share services and resources; key exchange mechanisms for node authorization and user authentication are provided to achieve a reliable communication. In a node joining step, the protocol uses identity cards, certificates, and hash SHA-1. The first node in the network creates the network and a casual session key. Symmetric encryption uses the AES algorithm to share session keys, being the execution times and energy consumption in cryptography procedures suitable for low-power devices. The asymmetric key encryption schemes ECC and RSA are used for device identification. Nodes must collaborate within Intranet or on the Internet. Access to WWW is available if one user has Internet connection. However, there could be more than one node for Internet access, where each node could share different services.

A complete self-configured secure light-weight protocol [22] for spontaneous wireless networks is presented. It uses a hybrid symmetric/asymmetric scheme and trust between users to exchange the session key and the keys to cypher the data. It is able to create the network and share secure services and resources to be used in devices with limited resources. The response times obtained are suitable to be used in real environments, and the storage and volatile memory required are quite low.

A self-configured secure protocol [23] for the creation of spontaneous ad hoc networks is presented. It uses a hybrid symmetric/asymmetric scheme and trust between users to exchange data. Secret keys are shared to encrypt the transmitting data. It may be used in devices with limited resources. The protocol allows to share resources and services in the network, and trust is achieved by only zero and first-level nodes. Certificates for every node that joins the network are obtained from a trusted node, and they are used to communicate with other nodes. A signature method is aimed at protecting against repudiation attack.

A secure self-configured protocol [24] to create and manage distributed and decentralized spontaneous network for data distribution and resources and services sharing among the users can be found. The protocol allows devices of different types to join and leave the network any time. A trust network can be built to obtain a distributed CA between the users that trust a new user. Asymmetric cryptography (RSA) and symmetric cryptography (AES) to exchange session keys are provided, each device has a public-private key pair for device identification, and there are no anonymous users. A summary SHA-1 is used to create a signature. Trust level, i.e., either trust or not trust, is established by looking physically, and it can change depending on node's behavior, even stop trusting. Simulation results have been obtained through NS-2 to validate the protocol, regarding packet delivery ratio, throughput, and average energy consumption.

EESCSP (energy efficiency secure protocol) [25] for self-configured spontaneous network creation and management is based on face to face trust establishment between joining an authenticating nodes, providing total security. It provides security while joining and accessing the services and resources into the network without Internet connection using trust level establishment mechanisms which will be secure, and the target is to nonexpert users. Users may join or leave the network. The protocol is aimed at saving the energy of nodes at the time of joining new node. RSA asymmetric encryption algorithm is used for authentication while AES symmetric key algorithm is used for communication and the session key to cypher messages. Certificates are created by summary SHA-1. The protocol can build a trusted network to obtain the distributed CA and uses laptops as mobile devices for creation and management performing integration automatically with little user intervention. An implementation in Java 1.6 or above on Windows 7 which creates a spontaneous wireless LAN using Wi-Fi technique instead of using Bluetooth among laptops is available. SPSNC (Secure Protocol for Spontaneous Network Creation) is compared with the proposed energy-efficient secure protocol (SPSNC-EE), regarding latency, packet hop count, and packet delivery rate resulting higher values for SPSNC-EE in all 3 metrics.

An enhanced distributed, lightweight, secure, and autonomous protocol [26] for the creation, communication, and management of spontaneous wireless ad hoc networks which uses a hybrid symmetric/asymmetric scheme and the trust between users in order to exchange the initial data and the secret keys is proposed. It is based on a social network imitating the behavior of human relationships, trust is based on the first visual contact between users, it is tailored for devices with limited resources, and it provides user-friendly security. A shared single secret key is used to create a communication crypto net to authenticate the holder as part of the secure group. The protocol allows to share resources and services securely. Asymmetric cryptography is provided in which each device has a public-private key pair for device identification, while symmetric cryptography is used to exchange session keys between nodes and encrypt the data using the shared session key. Identity cards are shared using cryptography algorithm. Malicious user revocation is also performed providing enhanced security. An implementation through NS-2 simulator is available, obtaining results in terms of normalized routing overhead, throughput, packet delivery ratio, and average delay, both for the protocol with revocation and without revocation, concluding that the protocol performs better with revocation method than without revocation.

A complete independent self-configured decentralized and distributed secure network creation protocol [8] for spontaneous wireless ad hoc networks which uses a hybrid

public-private key scheme and the trust between users in order to exchange the initial data and to exchange the secret keys that will be used to encrypt the data is presented. Network creation, communication, and management communication intrusion detection are available. It allows user-friendly operation, tailored for different devices and creating the network and sharing secure services and resources. The proposal is aimed at improving the communication and integration between different study centers of low-resource communities. Asymmetric cryptography (RSA) is used for device identification and authentication, and symmetric cryptography (AES) to share session keys between nodes. Authors focus on intrusion detection using signature detection technique to trace the intruders. Session key is revoked periodically to avoid network flooding. An implementation is available in order to test the protocol and compare with other spontaneous ad hoc network protocols. According to the results, the average delay in the proposal is better than that in regular architecture.

A complete self-configured secure protocol [27] for decentralized and distributed spontaneous wireless ad hoc networks which uses a hybrid public-private key scheme and the trust between users in order to exchange the initial data and to exchange the secret keys is presented. Its target is devices with limited memory space and limited energy. It allows to share secure services and resources and secured data distribution among authorized users in a user-friendly way. Symmetric key is used as session key to encrypt the message for AES, whereas an asymmetric key is used for user authentication and session key distribution process for RSA. Intrusion detection system is available for detecting different types of attacks, to protect the network, analyze, and find out intrusions. An implementation to test the protocol and compare with other spontaneous ad hoc network protocols. Minimal involvement of the user is required to configure the device mainly to establish trust. The protocol also performs session key revocation to avoid network flooding.

A complete self-configured lightweight secure protocol [28] for distributed and decentralized spontaneous wireless ad hoc networks uses a hybrid symmetric/asymmetric scheme, with little intervention from the user and the integration of different devices, in order to create and manage such networks and share data in devices with limited resources. A user without advanced technical knowledge can set up and participate in a spontaneous network, and the protocol provides user-friendliness. IP addresses identify each node, secure services are shared using TCP connections, and the network is built using IEEE 802.11b/g technology to share resources. Bluetooth allows authentication of nodes when they join the network. Session key revocation and intrusion detection mechanisms are also available. An implementation using J2ME and a fast virtual machine KVM is provided. The implementation of communication protocols is done over both Wi-Fi and Bluetooth. Crypto, i.e., a Bouncy Castle Lightweight API solution, has been selected since it provides a lightweight cryptographic open-source API. As for the results, the response times obtained are suitable to be used in real environments, whereas storage and volatile memory needs are quite low and the protocol can be used in regular resource-constrained devices.

The protocol in [29] creates a secure spontaneous ad hoc network used by different devices and allows the nodes to use the available services. After the creation, the nodes are clustered and a cluster head is assigned for each cluster. When a node in a cluster needs to access a service, a method is used to find and acquire the best available quality service from other nodes that have used service. These nodes provide information such as delay and transmission rate. Based on these information, trust value will be calculated for those nodes, and taking into account these trust values, the nodes that need service decide from which node the service must be accessed. When a node providing service moves to another cluster, service history management provides information about the migrated node. Node joining depends on the identity card, and the trust level is established by looking physically. A randomly created session key is distributed to all the nodes in the network. The service used in this proposal is file transmission. When the current cluster head's battery power level fails below a predetermined threshold or serves for a predetermined period of time, it broadcasts (within the cluster) a new election message. All the nodes then vote for a new cluster head, and the cluster head decides the winner based on simple majority. Experimental results have been obtained for comparison purposes. According to the percentage in which the nodes can access the quality service, the proposal is better than the existing system. In terms of the overhead vs number of nodes, the proposal has less overhead compared to the existing system.

In [30], two randomized neighbor discovery protocols for static multihop wireless ad hoc networks, known as CDH and CDPRR, which are based on collision detection, are presented. Simulations through Castalia 3.2 have been performed to compare both protocols against two randomized protocols from the literature, i.e., PRR [31] and Hello [5]. According to the results obtained through simulation, CDH and CDPRR outperform both Hello and PRR protocols in the presence of collisions regarding the neighbor discovery time, the number of discovered neighbors, the energy consumption, the throughput, and the number of discovered neighbors versus packet sent ratio, for both one-hop and multihop scenarios. As novelty compared to Hello and PRR protocols, both CDH and CDPRR are based on collision detection, know when to terminate the neighbor discovery process, and achieve to operate under more realistic assumptions. Furthermore, CDPRR presents better results in terms of time, energy consumption, and number of discovered neighbors versus packet sent ratio, while CDH does not need to know the number of nodes in the network.

3. Two-Phase Randomized Trusted Network Creation Model

In this section, we present a model for the creation of spontaneous wireless ad hoc networks based on trust.

3.1. Assumptions. For our proposal, we assume that each node can take a randomly chosen state either transmitting or listening; the nodes are static; each node has a unique identifier that distinguishes it from the others, e.g., MAC

address or manufacturer serial number; and the nodes are randomly deployed in a given area. Furthermore, time is slotted in rounds, the nodes require synchronization in slot boundaries, and the number of nodes N may be unknown by all the nodes in the network. In addition, each node is equipped with a limited range radio transceiver, and all the nodes have identical transmission range and can transmit or receive but not simultaneously, i.e., half-duplex operation. Moreover, each node has an internal memory to save local topology information such as neighbor identifiers, identity cards, and trust values. Furthermore, collisions may exist, the proposal allows a proper use both in one-hop and multihop environments, and the nodes can detect collisions and termination. We also assume that each device has its public-private key pair.

In Table 2, more in-depth information can be found about both protocols considered in this paper, that is, the reference protocol and our proposal.

3.2. Model. The proposed randomized protocol for the creation of spontaneous trusted networks consists of 2 phases. In the first phase, each node sends a BROADCAST packet towards the nodes within transmission range, containing its identity card, while in the second phase, each neighbor which receives the BROADCAST packet acknowledges sending a UNICAST packet, called ACK, which contains its identity card, towards the sender of the BROADCAST.

According to Figure 1, taking into account the existence of channel collisions, the time is slotted in rounds, and the slot width is τ. At the beginning of each round, every node randomly chooses a state either T (transmitting) with a probability p or L (listening) with probability $1 - p$. This state may be different among nodes in the same round and different among rounds in the same node.

We first introduce an example of the operation of the protocol in which all the nodes are within transmission range of all the others, i.e., one-hop scenario, shown in Figure 1. In the first round, nodes 1 and 3 transmit a BROADCAST packet, each packet represented by a red square, while the other node listens; thus, a collision takes place, and all the nodes continue contending in the following round. In round 2, only node 1 transmits a BROADCAST packet; thus, a successful transmission takes place and node 1 stops contending from now on; i.e., no red square appears in the following rounds for node 1. In round 3, the ACK sending (blue squares) from nodes 2 and 3 begin, and nodes 2 and 3 transmit the ACK thus provoking a collision and they both continue contending in the following round. In round 4, node 3 transmits successfully the ACK; thus, node 3 stops contending in the ACK sending. In round 5, all the nodes are listening; thus, node 2 continues contending in the ACK sending. In round 6, only node 2 sends the ACK successfully; thus, it will stop contending the ACK sending. At this moment, the ACK sending for node 1 ends. In round 7, node 3 transmits successfully the BROADCAST; thus, it will stop contending from now on. In round 8, the ACK sending for node 3 begins, and node 1 transmits successfully the ACK; thus, it stops contending in the ACK sending. In round 9, node 2 transmits successfully the ACK, stops contending the ACK sending, and finishes the ACK sending for node 3. In round 10, all the nodes are listening; thus, node 2 continues in the next round. In round 11, node 2 manages to transmit successfully the BROADCAST; thus, it will not contend from now on. In round 12, nodes 1 and 3 provoke a collision; thus, they both continue in the following round. In round 13, node 1 transmits successfully the ACK and stops contending. In round 14, node 3 transmits successfully the ACK and the algorithm finishes, since all the neighbors of every node have managed to acknowledge successfully and all the nodes have managed to transmit successfully.

Table 2: Qualitative comparison of the reference protocol and our proposal.

	[17]	Proposal
Number of phases	3	2
Mobile network	No	No
Randomized	No	Yes
Slotted time	No	Yes
N known	Yes	No
Requires synchronization	Yes	Yes
Requires a transmission schedule	Yes	No
Transmitting or listening (but not simultaneously)	Yes	Yes
One-hop	Yes	Yes
Multi-hop	Yes	Yes
Sleep available	No	No
Collisions considered	No	Yes
Collision loose transmission	Yes	Yes
Packet loss detection	No	No
Collision detection	No	Yes
Termination detection	No	Yes
Start transmission at different time instants	No	No
Uses hash checking	Yes	No
Uses signature checking	No	Yes
Discovers all neighbors	Yes	Yes
Succeeds at exchanging identity cards	Yes	Yes
Discovers all trusted neighbors	Yes	Yes

As shown in Figure 2, a flow diagram which highlights the operation of the protocol, in a round, right after choosing the state, each node in state T sends a single BROADCAST (identity_card) message (contend) in that round, corresponding to phase 1, and remains listening if the state is L or S, the latter meaning that it managed to transmit successfully in previous rounds. Notice that the identity card of each node contains the identifier, the public key, and the signature built using the private key (of the other fields).

Then, at the end of a round, the receivers have performed collision detection. A collision is a phenomenon that occurs when two or more nodes try to transmit simultaneously. Otherwise, we say that a node transmitted successfully, i.e., neither a collision nor idle slot takes place.

If the receivers detect that a node managed to transmit successfully the BROADCAST at the end of the round, these receivers send back a feedback packet simultaneously to the nodes within transmission range indicating this situation,

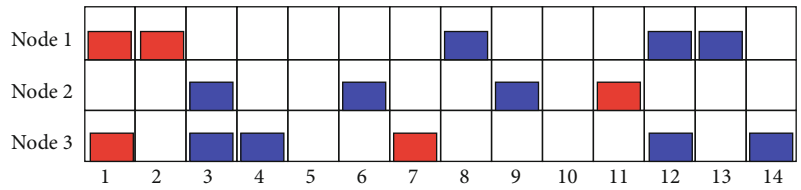

Figure 1: Proposal operation (timeline).

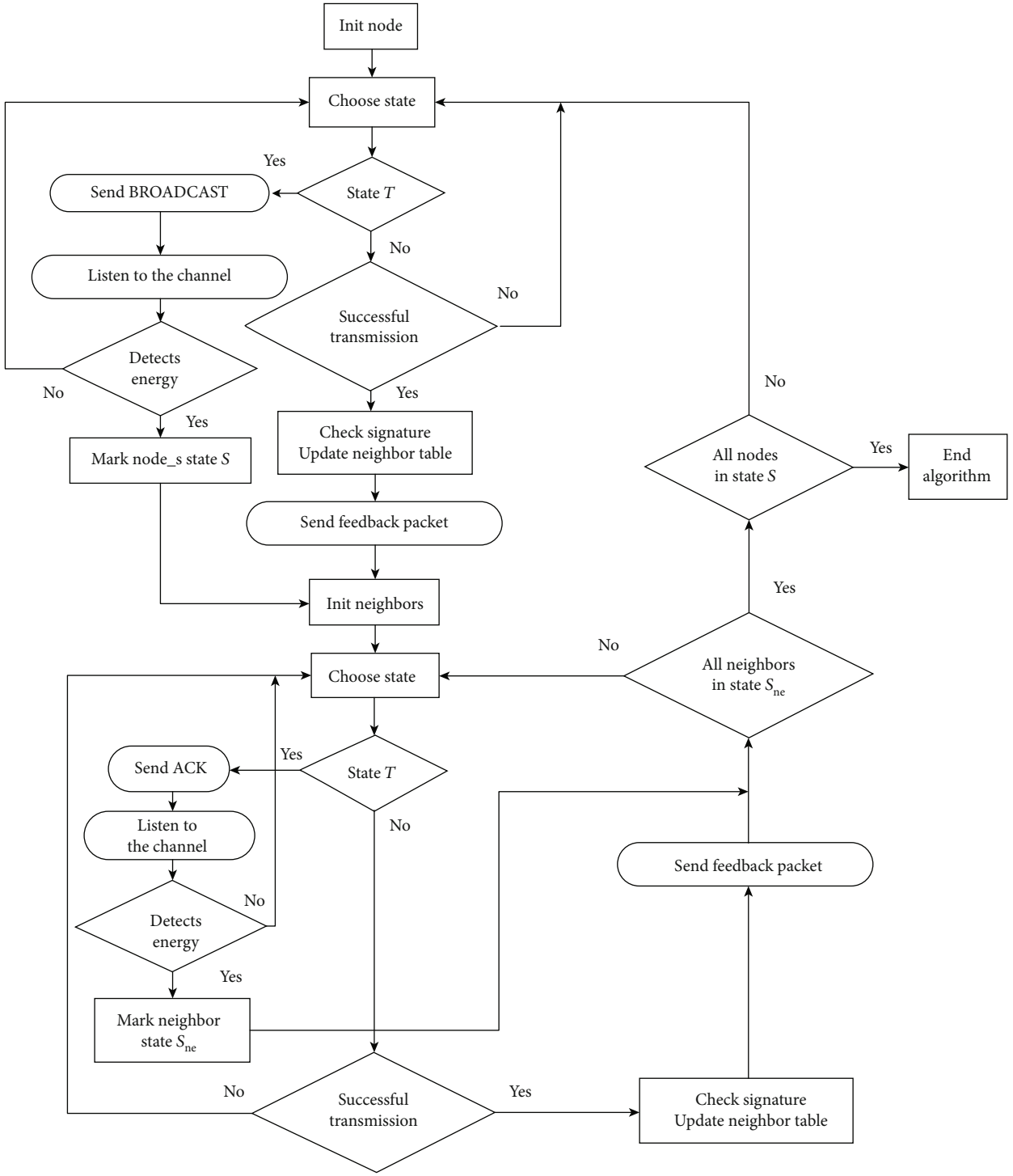

Figure 2: Proposal flow diagram.

in a broadcast manner. Otherwise, the receivers do not send any feedback packet, which indicates that a collision or idle slot occurred. Notice that the transmission of those feedback packets does not provoke a collision. At the same time, the nodes that transmitted the BROADCASTs listen to the channel and when energy is detected, the state of the node will change to *S*, meaning that the node transmitted successfully, and from that moment on the node in state *S* will remain listening, and the state *S* will not change for this node in the following rounds until the end of the algorithm. Notice that a node in state *S* will keep listening for incoming BROADCAST messages from other nodes, so that it can discover the other nodes within transmission range that do not collide, although it will keep sending the feedback packets when necessary. Also, notice that the feedback packet is much smaller than the BROADCAST packet, e.g., 802.11's ACK is 14 bytes.

Otherwise, i.e., no energy is detected, a collision or idle slot takes place, and all the nodes with state different from *S* will keep contending in the next round.

Moreover, if no signal is in the channel in a round meaning that no BROADCAST is received, that is, all the nodes are in state *L* or *S*, then the nodes in state *L* keep contending in the following round.

As soon as only one node transmits successfully a BROADCAST in a round, which we call node_s, a new process begins for the ACK sending (phase 2) in which all the neighbors, either in state *L* or *S*, i.e., the rest of the nodes within transmission range of node_s, will follow the same mechanism stated above sending ACK(identity_card) packets to node_s. In this case, we use S_{ne} instead of *S*, and at the beginning, all the neighbors have an initial state *I* different from S_{ne}.

When only a node sent successfully the BROADCAST, we say that this node transmitted successfully and the neighbors will save the node identifier and the identity card from the node in their neighbor tables. Furthermore, after reception of the BROADCAST, each neighbor checks the identity card signature (using the public key) and if it is okay, the sender is marked as trusted by the receivers. Otherwise, it is marked as valid. This trust value is also saved in the neighbor table. The private key must be kept in the node. If there is an error when checking the signature, this means that the message has been intercepted and manipulated somehow. Moreover, when a neighbor sends successfully the ACK, the same procedure is carried out in the other sense of communication. In this case, the neighbors will be trusted by node_s. Therefore, mutual trust could furthermore be established if necessary.

A termination detection mechanism is included in this algorithm. When all the neighbors acknowledged successfully, i.e., all the neighbors are in state S_{ne}, the current process for the ACK sending finishes and all the nodes begin a new round (shift to phase 1) and this process recurs. However, in this new round, the nodes that transmitted successfully before the ACK sending process began keep its state S.

The nodes know that all their neighbors are in state S_{ne}, i.e., the nodes know that the algorithm finishes for the ACK sending, when in several consecutive rounds (the number of consecutive rounds is a parameter that has to be carefully set), there are no transmissions detected; i.e., all the nodes within transmission range are in states *L* or S_{ne}. This procedure is valid since the probability that all the remaining nodes are in state *L* in a number of consecutive rounds is very low; thus, we conclude that all the nodes are in state S_{ne}. The same procedure is used to know that all the nodes are in state *S*; thus, the nodes know when the protocol finishes.

If all the neighbors acknowledged successfully for each node_s and not all the nodes transmitted successfully, a new round begins for the nodes (phase 1). If not all the neighbors acknowledged successfully, a new round begins for the ACK sending (phase 2). Otherwise, i.e., all the neighbors acknowledged successfully for each node node_s and all the nodes transmitted successfully; the algorithm finishes.

In this way, each node has a list of neighbors trusted or not (on base of their signatures) to conform the trusted network. The protocol solves the interception and the man in the middle problem but it does not solve the "sybil" node problem.

4. Simulation and Results

In this section, we assess the performance of the proposed protocol in comparison with a previous trusted network model [17]. For our experiments, we relied on the Castalia 3.2 simulator [9]. We only performed comparison to the protocol in [17] because, to the best of our knowledge, there is no other method apart from it available in the literature.

4.1. Overview of the Reference Protocol. As reference for our study, we have decided to use an existing trusted network creation protocol [17] that consists of 3 phases, that is, neighbor discovery, identity card sending, and identity card response, and is implemented in a deterministic way. The reference protocol allows a distributed public key management service through the proposal of a trusted model for the creation of spontaneous networks.

In the neighbor discovery phase, each node, one after another to avoid collisions according to a predetermined transmission schedule, sends 100 BROADCAST packets. After the arrival of all the BROADCAST packets to the nodes within transmission range, a threshold of 95% is set indicating the percentage of messages received above which a neighbor is considered as discovered.

Then, in the second phase, each node sends a PUBLICKEY message containing its identity card towards its neighbors, one after another to avoid collisions.

After the reception of a PUBLICKEY, in the third phase, each neighbor which received the PUBLICKEY message acknowledges, again one after another to avoid collisions, sending a PUBLICKEYRETURN message containing its identity card.

As soon as an acknowledgement is received, the node will save the neighbor identifier and the identity card from the neighbor in its neighbor's table. Furthermore, after reception of the PUBLICKEYRETURN, the node that sent the PUBLICKEY calculates the hash of the identity card and if it is equal to the hash received, the neighbor is marked as trusted, meaning that the node that sent the PUBLICKEY trusts the

neighbor. Otherwise, it is marked as valid. This trust value is also saved in the neighbor table.

When all the acknowledgments have been received, the next node sends the PUBLICKEY and the process recurs until the end of the algorithm, i.e., until all the identity cards have been exchanged and all the trusted neighbors have been discovered.

Moreover, all the neighbors in the network send acknowledgments according to a schedule even in the multihop scenario.

Table 2 includes a qualitative comparison of the reference protocol and our randomized trusted network creation proposal.

Among the most important features shown, we highlight the following: the reference protocol consists of 3 phases as stated above; it only works in static networks; i.e., it does not work in MANETs; and it requires synchronization and follows a predetermined transmission schedule. Although it considers collisions, it does not deal with them since it is a collision-free protocol, and it does not detect collision and termination conditions. The protocol will end when all the nodes and neighbors have transmitted successfully according to the schedule. Moreover, it manages to discover all the neighbors in the ideal case, succeeds at exchanging the identity cards, and discovers all the trusted neighbors. We reach to this conclusion since the protocol avoids collisions. In addition, it properly works both in one-hop and multihop networks, although the number of nodes in the network must be known to apply the schedule. However, the protocol is based on hash checking to discover the trusted neighbors.

To overcome the problems found in the existing reference protocol, we propose a randomized protocol. As shown in Table 2, the proposal consists of 2 phases, i.e., BROADCAST sending and ACK sending; it only requires synchronization in slot boundaries; it does not need a predetermined schedule; and the number of nodes may be unknown. The protocol can be properly used both in one-hop and multihop environments; it assumes that nodes provide collision and termination detection; thus, the nodes know when to terminate the algorithm and it manages to discover all the neighbors; and it succeeds at exchanging identity cards and discovers all the trusted neighbors. The protocol is handshake-based, and signature checking is used to discover the trusted neighbors. By introducing this proposal, we also aim to improve the performance in comparison to the previous reference protocol.

4.2. Simulation Setup. To obtain the results for both trusted network models, we used the same simulation scenario. For this purpose ,we varied the number of nodes in the network to test the performance of network scalability.

The simulation tool chosen for comparison purposes is Castalia 3.2 [9], which is based on OMNET++, and it is mainly used to test WSN (wireless sensor networks) and BAN (body area networks). In our case, it fully meets the requirements to validate trusted network protocols in static multihop spontaneous environments. For both protocols under test, we have set an identical $\tau = 0.07$ seconds, i.e., the time a node is transmitting using ZigBee radio.

We defined (i) a deployment area of 10 m × 10 m (one-hop, meaning that all the nodes are within the transmission range of all the others) and also (ii) an area of 100 m × 100 m (multihop setting meaning that only some nodes are within transmission range of the others), where N nodes are organized according to $M \times M$ grids.

For our experiments, taking into account the existence of collisions, the collision model parameter of Castalia has been set, which may take values 0 (no collisions), 1 (simplistic model for collisions), or 2 (additive interference model). In this case, we set the collision model parameter to 2 (i.e., the most realistic collision model).

The trusted network models use neighbor discovery techniques; therefore, for the simulations performed, we chose an output metric: the time consumption. Furthermore, both protocols manage to discover all their neighbors; thus, we will not present the results for the number of discovered neighbor's metric in this paper. However, the results for energy consumption, since the devices use batteries that may deplete in a given time; the throughput; and the number of discovered neighbors vs packet sent ratio have also been obtained through simulations for both protocols under test. We define the energy consumption as the average energy consumption of all the nodes. ZigBee takes into account the consumption when the radio is transmitting per node (i.e., 0.05742 Joules per second) or listening per node (i.e., 0.062 Joules per second). As for the throughput, we computed the number of packets received by every node, multiplied by the packet size, and divided by the time consumption. Finally, to obtain the number of discoveries vs packet sent ratio, we divided the number of discovered neighbors by the total number of packets sent by the nodes. The Castalia 3.2 simulator has an option available which shows the time and the average energy consumption and an option to show the number of packets sent and received.

For the experiments performed, we used a ZigBee (CC2420) radio model, setting a transmission power to 0 dBm, a packet rate of 5 packet/s, and the packet size to 2500 bytes.

For performance comparison, we set for our proposal different transmission probabilities: $1/N$, $1/2N$, $2/N$, and a fixed probability 0.25.

In Table 3, the simulation parameters are summarized.

4.3. Performance Results. In this section, we will focus on the simulation results comparing the performance of both target protocols under a one-hop setting and a multihop scenario.

4.3.1. Time Consumption. First, the results for a one-hop scenario according to the amount of time it takes the algorithms to create the spontaneous network based on trust will be presented. It is a simple case, although applicable to many real situations, especially when the radio transceiver technology has a very high transmission range.

As shown in Figure 3, our randomized proposal with transmission probability $2/N$ outperforms the reference protocol regarding time consumption in a one-hop setting for a number of nodes below 40. The randomized proposal with transmission probability $1/N$ also outperforms the reference

Table 3: Simulation parameters.

Parameter	Value
Static	True
Radio model	CC2420
Collision model	2
Transmission power	0 dBm
Packet rate	5 packet/s
Packet size	2500 bytes
Slot width	τ
τ	0.07 s
Size one hop	10 m × 10 m
Size multihop	100 m × 100 m
Deployment	Grid $M \times M$
Transm prob 1	$1/N$
Transm prob 2	$1/2N$
Transm prob 3	$2/N$
Transm prob 4	0.25

protocol in a network with less than 30 nodes. This improvement also takes place when the transmission probability is $1/2N$ for networks composed of less than 17 nodes. Overall, the randomized protocol with probability $2/N$ has the best performance while the proposal with a fixed probability of 0.25 is the worst regarding time consumption and the deterministic reference protocol presents intermediate results. As the network grows, there are more neighbors to be discovered and more identity cards to be exchanged; thus, the time consumption gets bigger. Mostly, the time consumption is worse in the deterministic protocol when the number of nodes is low due to the separate neighbor discovery phase which adds a lot of time consumption (i.e., each node sends 100 packets one after another to avoid collisions). Notice that the proposal and the reference protocol follow an increasing trend as the number of nodes increases.

In addition, we found that both protocols (i.e., the proposal and the reference protocol) achieve to discover all their neighbors, succeed at exchanging the identity cards, and discover all the trusted neighbors.

Next, we present the results obtained through simulation in a more realistic scenario: a multi-hop setting of size 100mx100m, i.e., a network in which only some nodes are within transmission range of the others.

According to Figure 4, our proposal with probability 0.25 outperforms the others, followed by the proposal with probabilities $2/N$, $1/N$, and $1/2N$. Finally, the deterministic protocol has clearly the worst performance. The protocols under test follow an increasing trend as the number of nodes grows, for the same reason stated above in a one-hop scenario. Again, the time consumption is worse in the deterministic protocol mostly since the additional neighbor discovery phase wastes a lot of time consumption. The schedule is the reason of this waste as the nodes transmit one after another to avoid collisions and most of the acknowledgments do not manage to reach their destination.

Again, we found that both protocols manage to discover all their neighbors, succeed at exchanging the identity cards, and discover all the trusted neighbors.

4.3.2. Energy Consumption. Regarding the energy consumption, as shown in Figure 5, the protocols under test present the same behavior as the time consumption for the one-hop case. To summarize, the randomized protocol with probability $2/N$ presents the best results while the proposal with a fixed probability of 0.25 is the worst regarding energy consumption and the deterministic protocols presents intermediate results. All the protocols under test follow an increasing trend with the number of nodes since as the time consumption increases, the energy consumption also gets bigger.

As shown in Figure 6, for the multihop case regarding the energy consumption, a similar behavior to the time consumption for the multihop case takes place; i.e., our proposal with probability 0.25 outperforms the others, followed by the proposal with probabilities $2/N$, $1/N$, and $1/2N$. Finally, the deterministic protocol consumes more energy than the other protocols. All the protocols under test follow an increasing trend as the number of nodes grows, for the same reason stated above in a one-hop scenario. The deterministic protocol spends more energy than the others since the additional neighbor discovery phase wastes a huge amount of time consumption. The schedule is the reason of this waste since the nodes transmit one after another to avoid collisions even in a multihop scenario and most of the acknowledgments do not reach their destination.

4.3.3. Throughput. According to Figure 7, which represents the one-hop case, the throughput is better for the reference protocol than for the proposal, both following a decreasing trend with the number of nodes. The proposal with transmission probability $1/N$ and 0.25 presents better results than the other probabilities for a number of nodes below 9. Overall, the proposal with probability $2/N$ outperforms the proposal with probability $1/N$, followed by the proposal with probability $1/2N$, and the proposal with probability 0.25 presents the worst performance in networks composed of more than 15 nodes. The decreasing behavior of the proposal is due to the decreasing of packets received per second since there are more collisions as the number of nodes increases. The deterministic protocol behaves better than the other solutions in a one-hop scenario since it is collisionless and all the packets sent are received.

Next, the throughput metric will be evaluated in a multihop setting and is shown in Figure 8. The randomized proposal with a fixed transmission probability 0.25 outperforms the other solutions, followed by $2/N$. The proposal with probability $1/N$ is better than the deterministic protocol and the proposal with probability $1/2N$ in networks composed of less than 32 nodes. The proposal with probability $1/2N$ is the worst for number of nodes above 20. To summarize, the proposal with probability 0.25 has the best performance while the proposal with probability $1/2N$ is the worst and the reference protocol has intermediate results. In addition, all the protocols under test present a decreasing

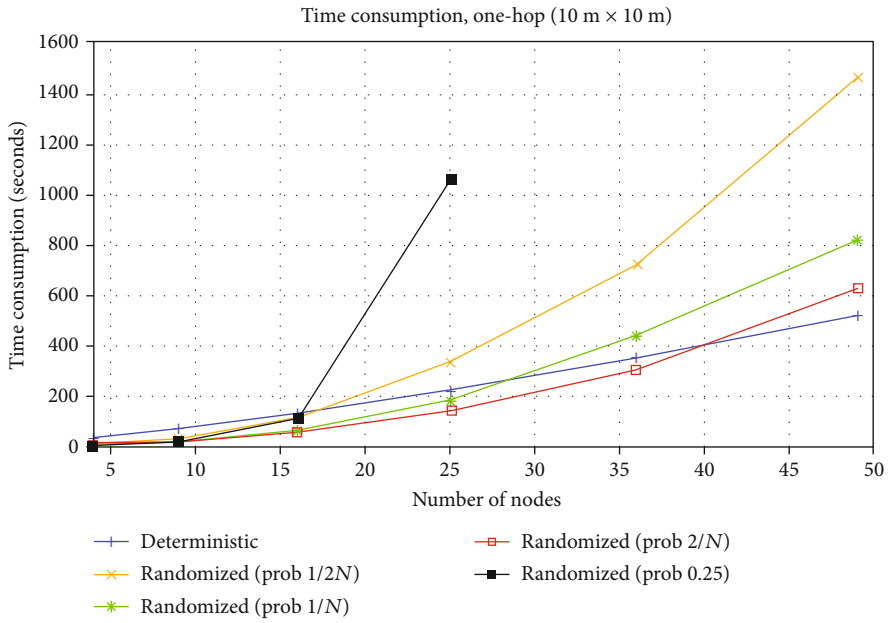

Figure 3: Time comparison; one-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm.

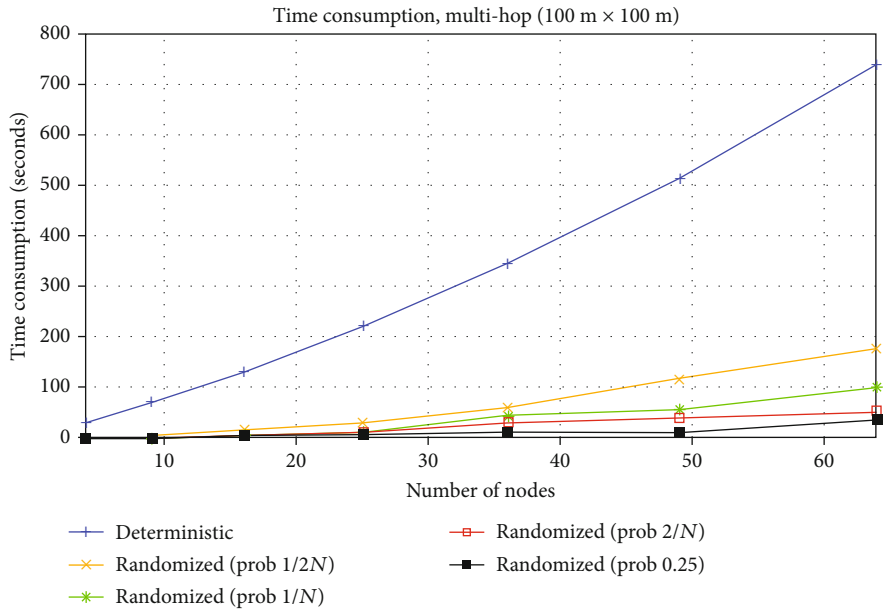

Figure 4: Time comparison; multi-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm.

trend as the number of nodes grows. The throughput in the deterministic protocol presents poor results especially with a low number of nodes since the time consumption is higher and the protocol spends a lot of time sending acknowledgments in the multihop case that do not reach their destination due to the schedule. Moreover, for networks of less than 10 nodes, the throughput is 0 byte/s since all the nodes are out of the transmission range of all the others and no packets are received.

4.3.4. Discoveries vs Packet Sent Ratio. In this section, the results regarding the number of discoveries vs packet sent ratio will be presented.

First, we focus on the results in a one-hop scenario.

According to Figure 9, our randomized proposal with transmission probability $1/2N$ outperforms the other solutions for number of nodes above 9, followed by the proposal with probability $1/N$, then the proposal with $2/N$, and finally the proposal with probability 0.25 for number of nodes below 15. Overall, the deterministic protocol presents the worst results with a constant ratio of approximately 0.008. This bad result is due to the number of transmitted packets in the neighbor discovery phase, which is above 100. Notice that in the neighbor discovery phase, each node transmits 100 packets one after another to avoid collisions. The protocols

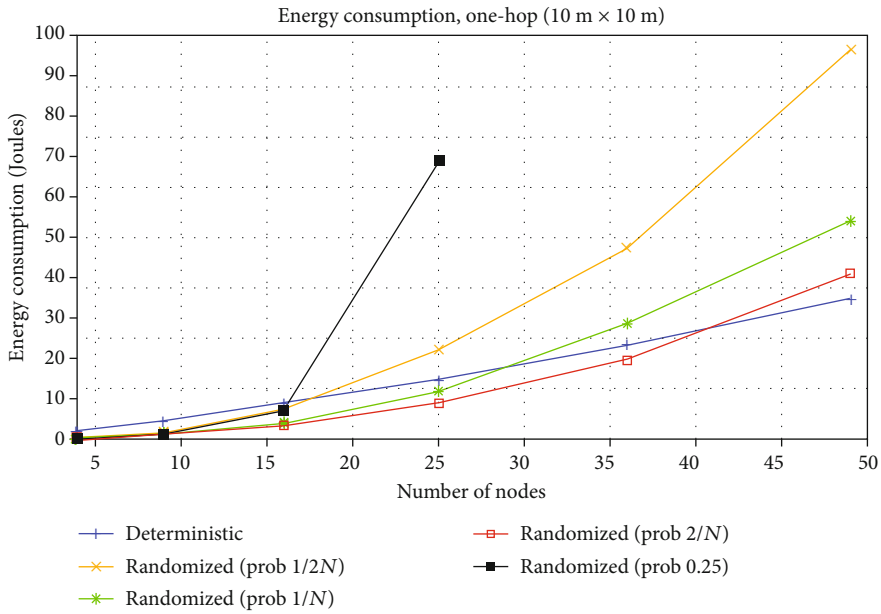

Figure 5: Average energy consumption comparison; one-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm; packet rate 5 packet/s; packet size 2500 bytes.

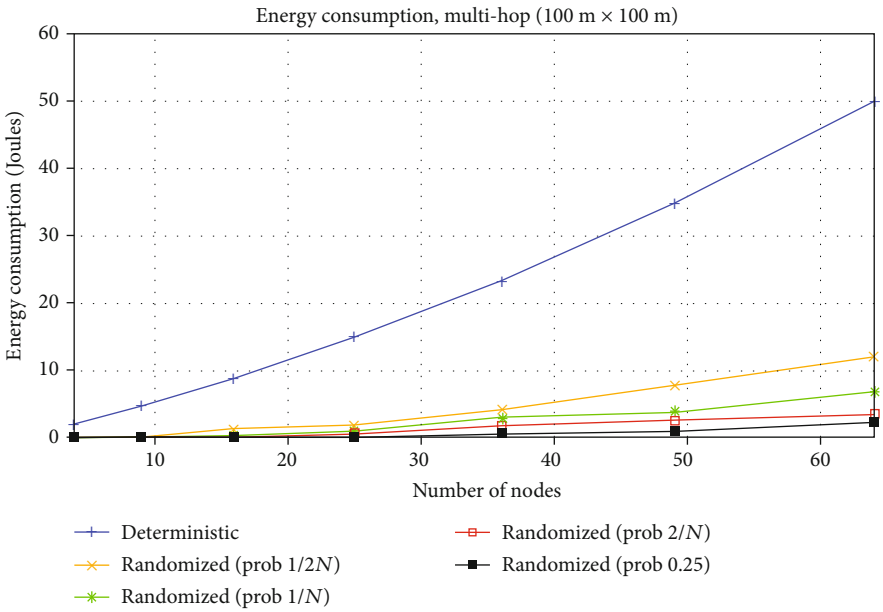

Figure 6: Average energy consumption comparison multi-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2;, transmission power 0 dBm; packet rate 5 packet/s; packet size 2500 bytes.

under test except for the deterministic protocol present a decreasing trend with the number of nodes.

As for the multihop network and as shown in Figure 10, the proposal with transmission probability $1/2N$ outperforms the others for number of nodes above 25, followed by the proposal with probability $1/N$ and $2/N$, and finally the proposal with probability 0.25. Again, the deterministic protocol presents the worst results, with a constant ratio of 0.002 . This bad behavior is again mostly due to the number of transmitted packets (i.e., above 100) in the separate neighbor discovery phase. The protocols under test except for the deterministic protocols also present a decreasing trend. The number of discoveries vs packet sent ratio for networks composed of less than 10 nodes presents a value of 0 since all the nodes are out of transmission range of the others and thus, the number of discovered neighbors is 0.

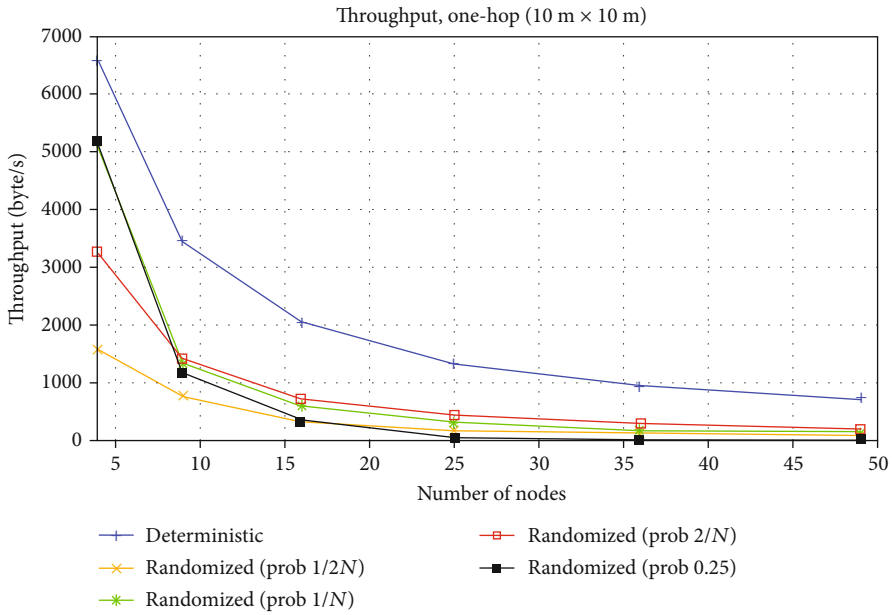

FIGURE 7: Average throughput comparison, one-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm; packet rate 5 packet/s; packet size 2500 bytes.

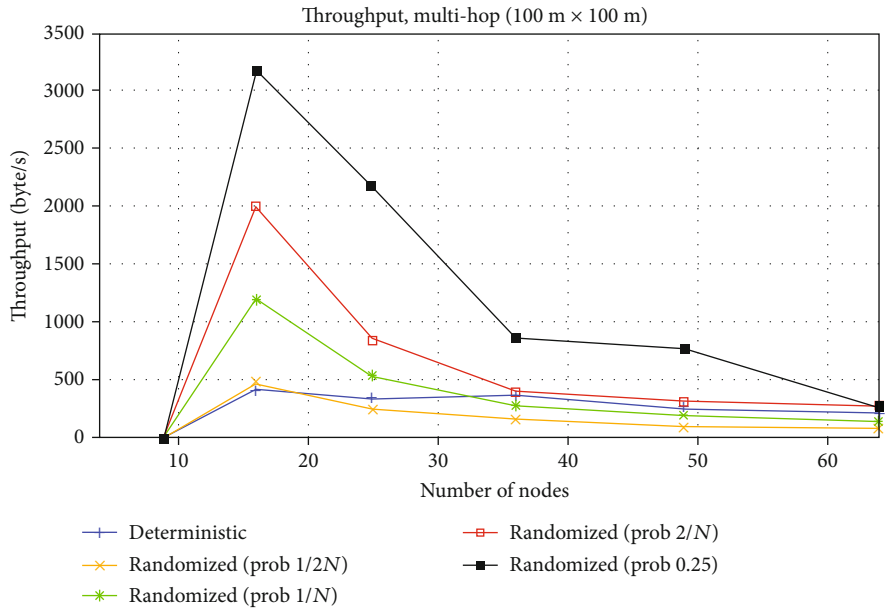

FIGURE 8: Average throughput comparison; multi-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm; packet rate 5 packet/s; packet size 2500 bytes.

5. Qualitative Comparison of Protocols

This section includes a qualitative comparison of protocols from the literature and our proposal, shown in Table 1.

According to the table, the articles present protocols for different types of networks, such as IoTs [16] for communities with low resources, mobile ad hoc cloud computing network [17], and spontaneous networks in [13, 18–23] and our proposal. However, all of them can be used in more general wireless ad hoc networks. Protocols in [13, 17, 19–23] are suitable to be used in mobile networks, while our proposal can only be used in static environments.

All the protocols present some common characteristics, such as they succeed at creating the spontaneous network, the devices have unique identities; they use identity cards, public-private key pair, and public key infrastructure; and they are designed for minimal user interaction. Moreover, they form a trust chain, and the trust value is based on

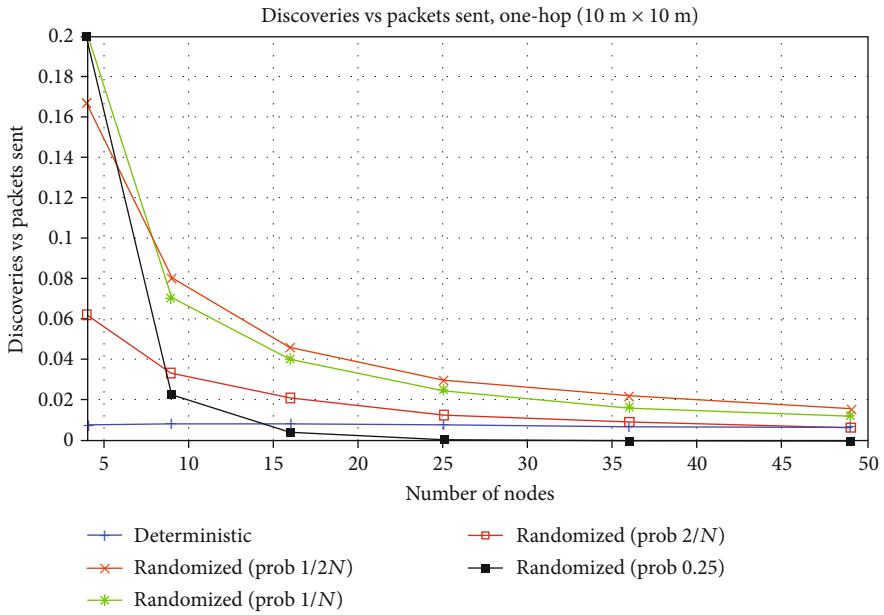

Figure 9: Average discoveries vs packet sent comparison; one-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm.

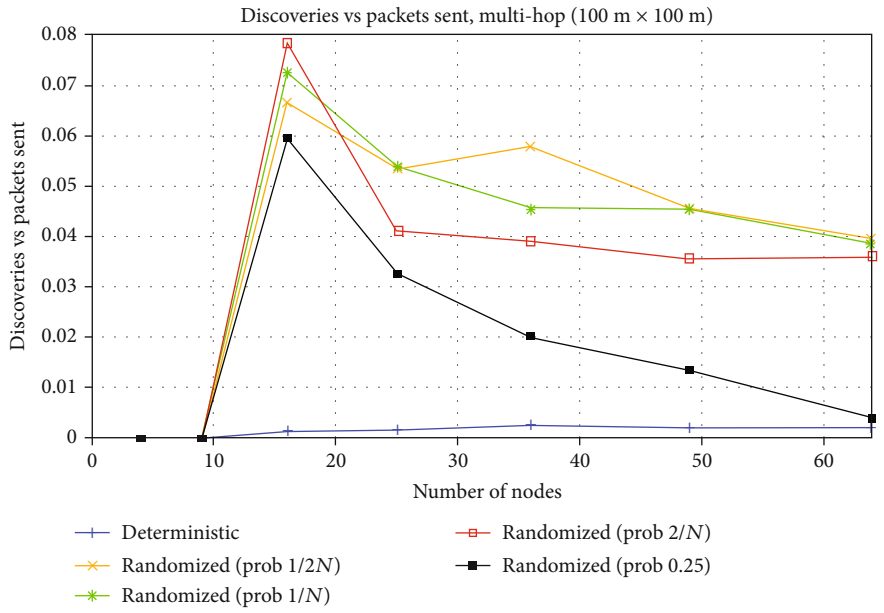

Figure 10: Average discoveries vs packet sent comparison; multi-hop setting; probabilities $1/N$, $1/2N$, $2/N$, and 0.25; collision model 2; transmission power 0 dBm.

human relations and includes a distributed CA. However, all the protocols except for our proposal allow nodes to join and leave the network at will. In our proposal, trust is established automatically and authentication is performed exchanging keys through ZigBee. Furthermore, our proposal is designed to allow network creation, while if new neighbors join the network, the protocol must be executed again.

Although most of the protocols in Table 1 include an encryption mechanism, our proposal does not consider an encryption procedure.

Authors in [18] developed a prototype in Java (J2ME) for mobile Nokia E65. Most of the protocols implemented for simulation use Castalia, some of them allow changing trust level, some use hash SHA-1, and most of them use a certificate.

The protocols presented in [16, 17, 20, 21] allow Internet access. Symmetric encryption AES and asymmetric encryption RSA/ECC are widely used; authentication through Bluetooth or ZigBee is also common. An intrusion detection technique is used in [19, 20] and [13], while [20] uses a caching technique.

6. Conclusion

In this work, we have carried out a study of trusted network creation strategies for spontaneous static multihop wireless ad hoc networks considering the presence of channel collisions. A novel randomized trusted network model has been proposed for static spontaneous network creation, which takes the advantages of collision detection, and no schedule is required for its operation. This model and an existing trusted network model used as reference have been implemented in the Castalia 3.2 simulator for comparison purposes.

The experiments have been focused on both one-hop and multihop environments, and four metrics have been chosen: the time consumption, the energy consumption, the throughput, and the number of discoveries vs packet sent ratio.

From the simulation results, we conclude that our novel proposal outperforms the existing protocol in terms of time and energy for low number of nodes and the number of discoveries vs packet sent ratio for a one-hop setting, while it outperforms the reference protocol regarding all four metrics in multihop scenarios. Furthermore, we assess the performance of our proposal setting the transmission probability to $1/2N$, $1/N$, $2/N$, and a fixed value 0.25. We also found through simulations that our proposal does not require to know the number of nodes, since a fixed transmission probability can be used, which provides reasonable results.

Moreover, our trusted network creation proposal allows collision and termination detection, it does not require a transmission schedule, and it consists only in two phases and follows more realistic assumptions.

In addition, we found that both protocols manage to discover all their neighbors, properly exchange their identity cards, and discover all trusted neighbors, in a quite reasonable amount of time and low energy consumption, in both one-hop and multihop scenarios.

The strategy used in our proposal to discover the trusted neighbors is that each node sends its identity card containing the signature built using the private key of the node. When the packet reaches the neighbor, it checks the signature using the public key of the node, and if it is okay, then the node is trusted by the neighbors. The same procedure is carried out in the other sense of communication; i.e., the neighbors are trusted by the node. Therefore, a mutual trust could furthermore be established if necessary.

Moreover, the computational complexity for our proposal, i.e., time consumption in both one-hop and multihop scenarios, is approximately $O(N^2)$ for transmission probability $1/N$, being N the number of nodes in the network.

The main practical limitations of our proposal is that it requires synchronization in slot boundaries and can only be used in static networks. To overcome these limitations, a synchronization mechanism must be used before the protocol begins, to allow the protocol to consider neighbors joining and leaving the network and nodes going in and out of each other's transmission range to be used in MANETs.

Among the practical applications, the proposal can be used in static spontaneous environments in a one-hop or multihop fashion. The number of nodes used to evaluate the proposal is low but enough for its application in real-world environments, such as in a sporadic meeting of students organized in a class or in a meeting between coworkers (i.e., a given location, to exchange information during a certain time) or even in a wireless sensor network deployed in the field to determine several parameters for watering services in a period of time or in a network of robots that exchange information with the aim of working together to fulfill a given task.

As possible future research work, we plan to develop and evaluate a new low energy consumption creation model based on trust for spontaneous wireless ad hoc networks and enhance the security mechanism to build a spontaneous network based on trust.

Appendix

In this appendix, we proceed to define the main key concepts used in this paper.

Spontaneous networks: a type of ad hoc network which is used in a certain location during a period of time, it does not depend on a central server, and the user is not required to be an expert, imitating human relationships in order to work together in groups, with minimal user intervention.

Trust: rely on a node to send information.

Trust relationship: established after identity card exchange and signature checking.

Trust chain: used to verify digital certificates.

Trusted neighbor: neighbor that is trusted by another node.

Identity card: structure exchanged that contains identifier, public key, and signature.

Public key: they can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures.

Private key: they must be kept secret, ensuring that only the owners of the private keys can decrypt content and create digital signatures.

Hash: summary function used to check integrity.

Signature: a way to secure the information and guarantee that the contents of a message have not been changed in transit.

Static network: the nodes cannot move in the deployment area, the nodes cannot get in and out of each other's transmission range, and neither node joining nor leaving the network is allowed.

Mobile networks (MANETs): the nodes can move throughout the deployment area and join or leave the network.

Deterministic protocol: the nodes need a transmission schedule and can discover all the neighbors with probability 1.

Randomized protocol: each node chooses a random time or state to transmit or receive and can discover all the neighbors with high probability.

Transmission probability: the probability that a node transmits.

Synchronization: coordination of simultaneous processes to complete a task according to an order.

Asynchronous: not synchronous.

Schedule: a list of planned activities to be done showing the times when they are intended to be done. In this paper, we refer to transmission schedule.

Collisions: take place when two or more nodes try to transmit simultaneously.

Collision models: used to model when a collision takes place.

Collision detection: process by which the collisions in the transmission are detected.

Termination detection: process by which the termination of the algorithm is detected.

One-hop network: all the nodes are within transmission range of all the others.

Multihop network: only some nodes are within transmission range of the others.

Neighbor: node within transmission range of another node, also called "one-hop neighbor."

Neighbor discovery: process by which the neighbors are discovered.

Neighbor table: local table in a node which contains neighbor identifiers and other information.

Certificate Authority (CA): an entity that issues digital certificates that certifies the ownership of a public key by the named subject of the certificate.

Time slot: a time during which something can happen or is planned to happen; in our case, transmit or receive.

Round: a time slot, i.e., portions in which the time is divided.

Handshake-based protocol: not one-way protocol, a handshake is an automated process of negotiation between two participants through the exchange of information.

BROADCAST: packet type that refers to packets that reach all the nodes within transmission range.

UNICAST: packet type that refers to packets that reach a given destination.

ACK: packet that acknowledges the reception of another packet.

PUBLICKEY: packet sent that contains the identity card.

PUBLICKEYRETURN: packet that acknowledges containing the identity card.

Feedback: acknowledgement.

Transmission range: maximum distance in which the packet can be received.

Contend: carried out when the nodes try to gain the use of the channel.

Successful transmission: a transmission that successfully reaches the destination, without collisions.

Identifier: information that identifies each node.

WSN: wireless sensor network.

BAN: body area network.

ZigBee (CC2420): specification of a set of high-level protocols of wireless communication for its use with low energy digital broadcast radio, based on IEEE 802.15.4.

Data Availability

The paper has no data available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been partially supported by the "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P. This work has also been partially supported by European Union through the ERANETMED (Euromediterranean Cooperation through ERANET joint activities and beyond) project ERANETMED3-227 SMARTWATIR.

References

[1] L. M. Feeney, B. Ahlgren, and A. Westerlund, "*Spontaneous networking: an application oriented approach to ad hoc networking*," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 176–181, 2001.

[2] G. Sun, F. Wu, X. Gao, G. Chen, and W. Wang, "Time-efficient protocols for neighbor discovery in wireless ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2780–2791, 2013.

[3] M. Conti, J. Crowcroft, G. Maselli, and G. Turi, "A modular cross-layer architecture for ad hoc networks," in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, J. Wu, Ed., pp. 1–12, Auerbach Publications, New York, NY, USA, 2005.

[4] A. Cornejo, S. Viqar, and J. Welch, "Reliable neighbor discovery for mobile ad hoc networks," *Ad Hoc Networks*, vol. 12, pp. 259–277, 2014.

[5] H. E. Ben, G. Chelius, A. Busson, and E. Fleury, "Neighbor discovery in multi-hop wireless networks: evaluation and dimensioning with interference considerations," *Discrete Mathematics and Theoretical Computer Science (DMTCS)*, vol. 10, no. 2, pp. 87–114, 2008.

[6] S. Preuß, C. H. Cap, and U. Rostock, "Overview of spontaneous networking-evolving concepts and technologies," *Rostocker Informatik-Berichte*, vol. 24, pp. 113–123, 2000.

[7] Z. Ye, T. Wen, Z. Liu, X. Song, and C. Fu, "*An Efficient Dynamic Trust Evaluation Model for Wireless Sensor Networks*," *Journal of Sensors*, vol. 2017, Article ID 7864671, 16 pages, 2017.

[8] N. Varghane and B. Kurade, "Secure protocol and signature based intrusion detection for spontaneous wireless ad hoc network," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, vol. 3, no. 5, pp. 758–768, 2014.

[9] A. Boulis, "Castalia - a simulator for wireless sensor networks and body area networks," *Version 3.2. User's Manual*, 2011, https://es.scribd.com/document/78901825/castalia-user-manual.

[10] M. Mani, A.-M. Nguyen, and N. Crespi, "SCOPE: a prototype for spontaneous P2P social networking," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 220–225, Mannheim, Germany, 2010.

[11] F. Legendre, M. D. de Amorim, and S. Fdida, "Implicit merging of overlapping spontaneous networks," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall*, pp. 3050–3054, Los Angeles, CA, USA, 2004.

[12] IBM, "A Smarter Planet," 2012, http://www.ibm.com/smarterplanet.

[13] P. Nimisha and M. P. Sindhu, "An enhanced secure protocol for spontaneous wireless ad-hoc networks," in *IOSR Journal of Computer Engineering (IOSR-JCE). International Conference on Emerging Trends in Engineering Management (ICETEM-2016)*, pp. 5–11, 2016.

[14] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless sensor networks and the Internet of Things: do we need a complete integration?," in *1st International workshop on the security of The internet of Things (SecIoT'10)*, Tokyo, Japan, 2010.

[15] D. S. Jadhav and D. A. Rokade, "A survey on security based spontaneous wireless ad hoc networks for communication based elliptical curve cryptography," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 11, pp. 3552–3555, 2014.

[16] R. Lacuesta, G. Palacios-Navarro, C. Cetina, L. Peñalver, and J. Lloret, "*Internet of things: where to be is to trust*," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 16 pages, 2012.

[17] R. Lacuesta, J. Lloret, S. Sendra, and L. Peñalver, "*Spontaneous ad hoc mobile cloud computing network*," *The Scientific World Journal*, vol. 2014, 19 pages, 2014.

[18] R. Lacuesta, J. Lloret, M. Garcia, and L. Peñalver, "A secure protocol for spontaneous wireless ad hoc networks creation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 629–641, 2013.

[19] S. Jadhav, P. Naik, and K. Kahade, "A survey on security based on user trust in spontaneous wireless ad hoc network creation," *International Journal for Innovative Research in Science and Technology*, vol. 2, no. 10, pp. 84–88,, 2016.

[20] P. M. Nandagawli, A. R. Tayal, and A. Jaiswal, "A survey on symmetric key protocol for spontaneous wireless ad hoc network creation," *International Journal of Scientific and Technology Research*, vol. 3, pp. 89–91, 2014.

[21] N. S. Reddy and J. G. Ponsam, "*Security based on user trust in spontaneous wireless ad hoc network creation*," *International Journal of Engineering Development and Research*, vol. 2, no. 2, pp. 1473–1480, 2014.

[22] N. M. V. Satyannarayana and M. R. Veerababu, "*A secure protocol for spontaneous ad-hoc networks*," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 7, no. 6, pp. 2502–2506, 2016.

[23] J. B. Kallada, "A protocol for creating spontaneous ad hoc wireless network for secure communication," *International Journal of Mechanical Engineering and Information Technology*, vol. 3, no. 3, pp. 1061–1066, 2015.

[24] K. V. Shinde, H. Kaur, and P. Patil, "Enhance security for spontaneous wireless ad hoc network creation," in *2015 International Conference on Computing Communication Control and Automation*, pp. 247–250, Pune, India, 2015.

[25] D. N. Rewadkar and S. B. Karve, "Energy efficient self configured secure protocol (EESCSP) for wireless spontaneous ad-hoc network," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 792–799, Kanyakumari, India, 2014.

[26] R. Firoj and A. Antil, "Jain Enhanced security protocol for spontaneous wireless ad-hoc network," *International Journal Of Engineering And Computer Science*, vol. 5, no. 12, pp. 19419–19428, 2016.

[27] N. Varghane, B. Kurade, and C. Pote, "Intrusion detection, secure protocol and network creation for spontaneous wireless ad hoc network," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 2, pp. 389–394, 2014.

[28] K. Srinivas, G. B. NarasimhaRao, and S. S. Reddy, "A self-configured secure protocol for the management of wireless ad hoc networks," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5529–5532, 2014.

[29] G. Pradeep and A. P. Shobak, "Improving QoS in spontaneous ad hoc neworks," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5705–5707, 2014.

[30] J. V. Sorribes, L. Peñalver, C. T. Calafate, and J. Lloret, "Randomized neighbor discovery protocols with collision detection for static multi-hop wireless ad hoc networks," *Telecommunication Systems*, 2021.

[31] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and fexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM international symposium on mobile ad hoc networking computing, ACM Press*, pp. 137–145, 2001.