Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 232083, 16 pages
doi:10.1155/2010/232083

*Research Article*

# A Spontaneous Ad Hoc Network to Share WWW Access

**Raquel Lacuesta,[1] Jaime Lloret,[2] Miguel Garcia,[2] and Lourdes Peñalver[2]**

[1] *Universidad de Zaragoza, Ciudad Escolar s/n, 44003 Teruel, Spain*
[2] *Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain*

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

Received 15 September 2009; Revised 27 April 2010; Accepted 31 May 2010

Academic Editor: Weihua Zhuang

In this paper, we propose a secure spontaneous ad-hoc network, based on direct peer-to-peer interaction, to grant a quick, easy, and secure access to the users to surf the Web. The paper shows the description of our proposal, the procedure of the nodes involved in the system, the security algorithms implemented, and the designed messages. We have taken into account the security and its performance. Although some people have defined and described the main features of spontaneous ad-hoc networks, nobody has published any design and simulation until today. Spontaneous networking will enable a more natural form of wireless computing when people physically meet in the real world. We also validate the success of our proposal through several simulations and comparisons with a regular architecture, taking into account the optimization of the resources of the devices. Finally, we compare our proposal with other caching techniques published in the related literature. The proposal has been developed with the main objective of improving the communication and integration between different study centers of low-resource communities. That is, it lets communicate spontaneous networks, which are working collaboratively and which have been created on different physical places.

## 1. Introduction

A spontaneous ad hoc network is type of ad hoc network that is formed in a certain time during a period of time, with no dependence on a central server and without the intervention of an expert user, in order to solve a problem or carry out a specific task [1]. This network is built by several independent nodes coming together at the same time and in the same place to be able to communicate with each other. Nodes are free to enter and leave the network and they could be mobile or not. Spontaneous networking happens when neighboring nodes discover each other within a short period of time; however, the velocity of discovery is paid in terms of energy consumption [2]. Spontaneous networks are conceptually in a higher level of abstraction than ad hoc ones; they are basically those which seek to imitate human relationships in order to work together in groups, running on an existing technology. Their objective is the integration of services and devices in an environment which allows the provision to the user of an instant service with minimum manual intervention. The concept of spontaneous networks was introduced in depth by Laura Marie et al. in the paper in [3].

The main features in spontaneous networks are the following.

(i) Network boundaries are poorly defined.

(ii) The network is not planned.

(iii) Hosts are not preconfigured.

(iv) There are not any central servers.

(v) Users are not experts.

In this type of network the configuration services needed depend mainly on the network size, the nature of the participating nodes, and the applications that have to be carried out.

Latvakoski et al. proposed a communication architecture for spontaneous systems in [4] which integrates application-level spontaneous group communication and ad hoc networking together. Zarate Silva et al. proposed AWISPA [5], a collaborative learning environment based on wireless spontaneous networks.

One of the main issues that difference the spontaneous networks from other fixed or mobile networks is that they facilitate the integration of services and devices, setting up both the new services and the configuration parameters of the devices. It has to be done without the user intervention or interference in the operation of the network. The malfunction or failure of one of the devices or services does not compromise the viability of the community. Any resources being used by the community which malfunction are automatically released and the service is deregistered.

A spontaneous network enables a group of devices to work together collaboratively while they are located very close to each other with a minimum interaction. It can be used for sharing resources and internet services. But, we should take into account the limitation of the resources of the devices. Just one of the nodes has to be connected to Internet to share its connection and its resources to the whole network. Caching techniques are demanded in order to avoid the overload of the nodes. Moreover, configuration with a minimal interaction from the user and security on the communication should be established. There are many application areas for ad hoc spontaneous networks: industrial (communication between sensors, robots, and digital networks), business (meeting, stock control, etc.), military (hard and hostile environments), and teaching. The range of environments in which these networks can be applied is wide and may include conference services and other "ubiquitous computing" applications at home or office.

This paper shows the design and simulation of a model that lets optimal spontaneous network access using a caching mechanism. We present the procedure of the nodes involved in the system, the security algorithms implemented, and the designed messages. Moreover, we included the analytical proposal and its comparative with the most similar protocols in the literature. The validation of the protocol is carried out through several simulations and comparisons with regular architectures. The proposal has been developed with the main objective of improving the communication and integration between different study centers of low-resource communities.

The paper is structured as follows. The model proposed to create spontaneous networks is presented in Section 2. Section 3 shows the auto-configuration procedures used in the proposal. Section 4 analyzes the model analytically. The security is discussed in Section 5. The protocol procedure and the messages designed are shown in Section 6. The comparison of our proposal with some caching techniques is presented in Section 7. In Section 8, the model is validated through the simulation and comparison with a regular architecture. Finally, Section 9 summarizes this work and points out the main conclusions.

## 2. Spontaneous Network Proposal Description

If some people wish to build a spontaneous network, they may meet in a physical space at a given moment in order to make use of services such as group communication, cooperation on running programs, security, and so forth. The members who make up this community may vary at any specific time (users may join or leave at will).

When a device joins the network, it must follow the following steps.

(1) *Integration the Device into the Network.*

   (a) Agree the transmission protocol and speed.
   (b) Configure node addresses, routing information and other resources.

(2) *Discovery of the Services and Resources Offered by the Devices.*

   (a) Discover the services and resources shared in the network.
   (b) Have a list of services and resources available in the network updated.

(3) *Access to the Services Offered by the Devices.*

   (a) Manage the automatic integration tasks and the use of, for example, agent service.
   (b) Manage access security to the services.
   (c) Manage the join and the leave of nodes of the network.

(4) *Collaborative Tasks.*

   (a) Within the intranet, among the various members.
   (b) On the internet, with the other communities.

We emphasize that the main difference with ad hoc networks is that spontaneous networks are generated to work during a period of time on a limited space. Spontaneous networks are user-oriented and application-oriented networks that are based on human relationship and take into account the security and performance. A quick creation and configuration of these networks will be fundamental to their performance. In [6], Feeney et al. explain the difference between ad hoc and spontaneous networks and, moreover, they identify five key challenges posed by the spontaneous networking environment. In our proposal we follow this because the devices have a similar behavior to human relationships. It lets a minimal intervention of the users and a quick configuration of the network and its security.

Many routing protocols for Mobile Ad hoc NETworks (MANET) such as Destination-Sequenced Distance Vector (DSDV) [7], Dynamic Source Routing (DSR) [8], Ad hoc On Demand Distance Vector (AODV) [9], and Temporally-Ordered Routing Algorithm (TORA) [10] could be used in spontaneous networks. These protocols work with the concept of route discovery to locate the packet's receiver. In some cases, the protocols use caching methods to avoid looking for a route each time data have to be transmitted. We use this idea in spontaneous networks to improve the overload of the nodes, specially of those that act as gateways of the network.

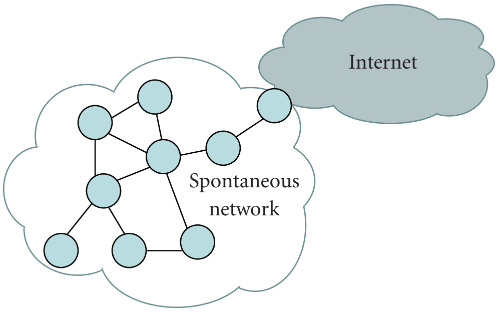

Figure 1: Proposed model.

## 3. Network Autoconfiguration Proposal

When the devices join a network, they must have to be aware of all the different tasks needed to communicate with each other and the configuration of both logical and physical parameters (all of them should be automatic) [11].

Users bring their resources to the system. If one of the users of the spontaneous network has Internet connection with WWW access, the connection will be shared and that device will be the one that provides the access to the WWW (There could be applied to other services such as e-mail, filesharing, etc.). It could be more than one Internet access in the spontaneous network and each one of them could share different services. In this model a user contributes capabilities, technical resources to access external services, and other applications (reports, games, and other data which they may wish to share). The intranet and its view to the outside world permits the community both internal and external cooperation. Figure 1 shows the model example.

The resources of the devices can also be used according to their available capacities. One user may be responsible for processing a specific task if another one needs to carry this out but does not find this possible owing to the fact that the device does not have the enough resources.

The following tasks should be performed when a user joins the spontaneous network:

(1) node identification,

(2) identification between nodes,

(3) address assignment,

(4) to join services.

These tasks should be carried out with security. Consequently, when configuring an ad hoc network, one of the main problems which arise is the generation of a unique IP address. Most of the routing protocols assume that the mobile nodes are configured a priori with a unique IP address before becoming part of the network, which is not here the case. The problem comes from not knowing the topology of the network, neither when being set up, nor later on modification. A node may enter or leave the network at will at any time, so a protocol must be capable of managing the generation of these IP addresses in order to run the network properly [12]. Also, the protocol must be able to detect the existence of duplicated IP addresses, which may occur, for example, when two subnetworks join together, or when a node which leaves one subnetwork with an IP (until then unique) joins another, or even when there is a substitution attack on the nodes.

Some authors have solved this problem using DHT-based algorithms [13] as self-organizing systems and others use hypercubes to implement indirect routing [14]. Faced with this challenge and after analyzing the working of ad hoc networks, within the framework we have set ourselves, we propose a distributed and decentralized solution.

Our proposal begins with the awareness that ad hoc spontaneous networks need a flexible protocol which adapts itself to any number of different nodes and to their various characteristics. In the formation of these networks a range of different devices (cell phones, PDAs, laptops, etc.) may take part. These nodes have to be configured in order to be part of the network. Despite the fact that our networks do not include central servers, the operating of the wireless network must be similar to the one with IP configuration infrastructure: translation of DNSs, service identification, etc. On the other hand, it is required the minimum intervention of the user because it will be used by nonexpert users, so the configuration must take place independently. The configuration of all the parameters necessary to form such networks implies to exchange information between the nodes. In our proposal, the node's IP addresses configuration has two main phases: first, a local connection address is generated by the node which wants to form part of the network. In order to generate the address we fixed the network identifier to a class B network that starts with 169.254. The rest of the IP is formed by the chain of a random number of four bits, which lets regenerate the IP if it has been duplicated, and twelve bits obtained from the twelve last bits of the obtained hash when we pass a hash function to the user's data. Second, we must check the IP duplication by one of the nodes that is already in the network. In order to perform this check the node uses a broadcast technique that sends a packet with the proposed IP. If a node is using this IP, it responds to the new node. As the IP cannot be used by the new node, it has to propose a new IP. More details about the automatic configuration procedure can be read in the paper in [15].

Our approach is based on human relations. The set-up configuration is based on presentation or greeting. In a group of friends, a new individual is introduced to the other members by one of the participants of the meeting. This member already knows the other's presentation data or may obtain them at the moment of presentation. He or she is then responsible for facilitating the new member's integration easily and simply into the group. Consequently, the network management is formed and run by cooperation between nodes, behaving similarly to that in human relations existent in our society. Thus, the formation of these networks is carried out in two principal phases: the first one is the presentation, greeting, or preidentification, and the second phase deals with the creation of the network and communication. As we can see in [16], the social relationship could be modeled as a spontaneous network. This has been the main reason to make a communication network based in this type of communications.

The presentation phase follows the human rituals enacted when different individuals come together to form a work group. This is carried out by one of the nodes already belonging to the network. In this phase, the devices exchange the necessary information in order to be recognized; by presenting this information they gain access to the network. In this manner, any user may come to be part of the network without having high level of computing knowledge. The user connected to the device has to input his or her personal information when access the network for the first time. Automatically, a data configuration proposal is generated and the devices available within range are identified. The intervention of the user is limited to select the user among those detected by the device and with which he or she wished to preauthenticate. Once the device has been selected, the interexchange of presentation information can take place automatically between the two nodes; this information, after being exchanged and authenticated, allows the participants to gain access to the generated network.

In this network, each node acts both as client and as server, sending and receiving information and providing services to the other nodes in the network on request. The devices which make this up may be PDAs, laptops, cell phones, among others.

We have developed our proposal for devices with limited resources to allow them to surf the web and share the files and resources inside the spontaneous network. Although it could be extended to other internet services, because of their limited resources we have restricted to just transport http. The protocol runs in the devices as a process with no restrictions. It is not centrally controlled, does not overload any single network node, and does not need a central authority to start up or manage the insertion of new nodes.

## 4. Analytical Model

Given the inherent characteristics of a spontaneous network, the network depends on the number of devices, their position, and the number of connections between the devices of the network. The nodes could be stationary, because their physical placement does not vary over the time, or mobile, and they are placed in the network for a limited period of time. In our model the devices could leave the network voluntarily or because of energy constraints, so we will not take into account the energy of the devices in our analytical model because the spontaneous network may be disappear before the energy of the ad hoc devices is consumed.

We assume the spontaneous network lifetime limited and divide the total time into individual time periods, represented by $t \in T$ for $t = 0, 1, \dots, t_z$. At time $t_z$, the last node leaves the network. We will use the graph theory to define the network. Let $G = (V, S, P, E)$ be a spontaneous network in the time $T$, where $V$ is a set of devices $v_i$ with $i = \{0, 1, \dots, n\}$ and $n = |V|$ (number of devices in whole network), $S$ is a set of services offered by the devices of the network, $P$ is the placement function assigned to every element of $V$, (the placement function assigns to every device $i$ of $V$ and to any time $t$ a set of coordinates $P_i = [x_i, y_i, z_i]$) and $E$ is a set of their connections. The cardinality of $V$ changes along the time because nodes can join or leave the network at will; if the device is mobile, its position $P$ changes along the time and the cardinality of $E$ changes with the creation and deletion of connections.

We can define $N(v_i)$ as the neighbourhood of node $v_i$ as it is shown in

$$N(v_i) = \bigcup_{v_j \in V,\, v_j \neq v_i} \left\{ v_j \mid E\left(v_i, v_j\right) < tx_{\text{range}_i} \right\}, \qquad (1)$$

where $tx_{\text{range\_}i}$ is the transmission range of $v_i$. The neighbourhood of a node is the set of nodes which is within its transmission range. Connections allow two-way communication (bidirectional links), so that connected nodes can communicate with each other in either direction, that is, $E(v_i, v_j) = E(v_j, v_i)$.

Let us suppose that at time $t = t_0$ a device enters the radio coverage area of another device and, therefore, they become neighbours. Let $R$ be the random variable that represents the time when a new node appears in the network. Responsiveness of the discovery process, $F_R(t)$, can be represented by the probability function of the random variable R, which is defined as it is shown in

$$F_R(t) = P\{R \leq t\}, \quad \forall t \geq t_0. \qquad (2)$$

Now, let us know when the devices of the network will discover a service provided by the new device. In order to calculate it, Figure 2 is provided. It shows the reference times used in our analytical model. First, devices of the network send a service search to the network. Then, the new device joins the network at time $t_0$. The device announces its service $t_1$ seconds after it has joined the network. The new device will leave the network $t_2$ seconds after it has joined the network. The next service search of the nodes of the network will be $t_3$ seconds after the node has joined the network. So, $t_3$ should be lower than $t_2$ in order to provide its services to the network.

The service probability, $S(v_i)$, is defined as the probability that a device $v_i$ has to hear any browsing requests and to announce its service during its appearance in the network. Using the information shown in Figure 2, $S(v_i)$ is defined in

$$S(v_i) = \Pr(t_2 > t_3)\Pr(t_2 > t_1) = \int_0^{t_z} \Pr(t_2 > t_3)\Pr(t_2 > t_1)\, dt. \qquad (3)$$

The number of nodes in the spontaneous network is $n$. For simplicity, we will suppose that a new node announces a service, and the other nodes browse the service. On the other hand, for this case, we assume that the service announcing and browsing interval follows an exponential distribution with rate $\lambda$ (proportional to $-\lambda \cdot e^{-\lambda t}$) and the node residence time follows an exponential distribution with

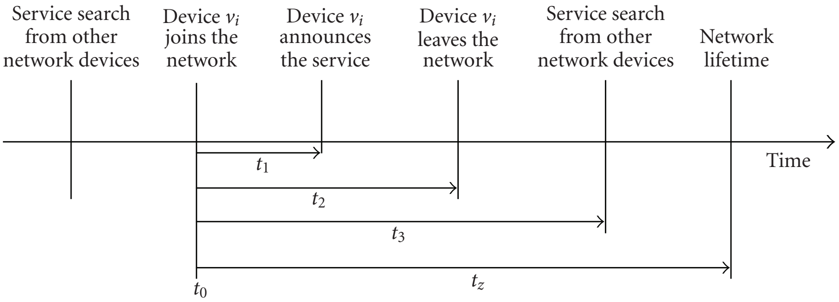
Service search from other network devices
Device $v_i$ joins the network
Device $v_i$ announces the service
Device $v_i$ leaves the network
Service search from other network devices
Network lifetime
Time
$t_1$
$t_2$
$t_3$
$t_z$
$t_0$

Figure 2: Reference times when a device $v_i$ joins and leaves the spontaneous network.

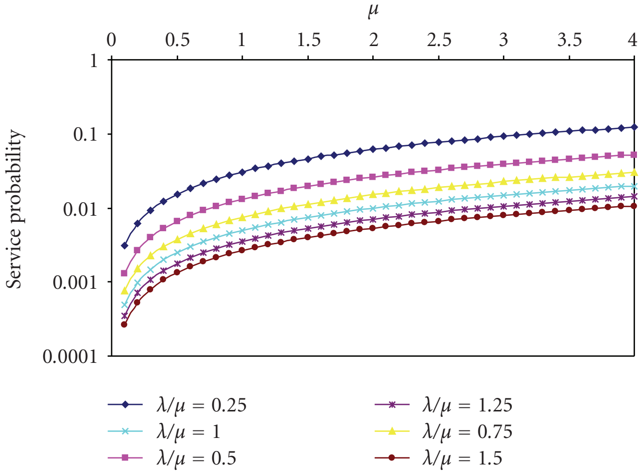

Figure 3: Service probability in the spontaneous network.

rate $\mu$ (proportional to $-\mu \cdot e^{-\mu t}$). Now, the service probability can be expressed as it is shown in

$$S(v_i) = \left(\frac{\mu}{n \cdot \lambda + \mu}\right) \cdot \left(\frac{\mu}{\lambda + \mu}\right) = \frac{\mu}{(n \cdot (\lambda/\mu) + 1) \cdot (\lambda/\mu + 1)}. \tag{4}$$

In order to show the service probability graph, we have used several values for the relationship $\lambda/\mu$ in a network of 100 nodes as a function of $\mu$ values. Figure 3 shows the service probability obtained. A node offers higher service probability if $\lambda/\mu$ has lower values. So, the node residence time ($\mu$) has to be higher than the service announcing and browsing interval ($\lambda$). On the other hand, although the service probability increases as $\mu$ increases, we have to take into account that less number of nodes in the spontaneous network will give higher service probability values.

On the other hand, we denote a random variable $t_x$ to be the service time. If the service is offered until the node leaves the network, $t_x = t_2 - t_1$. Let $g$ be the number of time slots required to transmit a message if the node is within the service area. With constant message length $L$ and fixed bandwidth $w$, we have $g = L/w$. In each time slot, a node has the probability of $S(v_i)$ to receive the service announcement. Thus, the distribution function of $t_x$, that is, the probability that the message can be transmitted within no more than $b$ time slots, is a random variable with Pascal distribution [17]. It can be expressed by

$$F_{t_x}(x) = \sum_{i=0}^{b-g} \binom{g+i-1}{g-1} S^g(v_i)(1 - S(v_i))^i. \tag{5}$$

This function has a mean value of $g/S(v_i)$ and a variation of $(s \cdot (1 - S(v_i)))/S^2(v_i)$.

Assuming that the devices are able to keep a maximum value of $k$ messages in its queue, the data generation and transmission can be modeled as an M/G/1/k queue. Since the activation period is exponentially distributed, the message arrival is a Poisson process with an average arrival rate of $\lambda$. The service rate, $\mu$, depends on the available bandwidth $w$ between nodes, the message length $L$, and the service probability $S(v_i)$ as it is shown in

$$\mu = \frac{w \cdot S(v_i)}{L}. \tag{6}$$

Finally, we derive the steady-state probabilities of the M/G/1/k queue in order to know the probability of $q$ arrivals ($k_q$) during the period for serving a message. According to the Poisson distribution of message arrival, we obtain

$$k_q = \sum_{t=g}^{t_x} \frac{e^{-\lambda t}(\lambda t)^q}{q!} \cdot \binom{t-1}{g-1} S^g(v_i)(1 - S(v_i))^{t-g}. \tag{7}$$

## 5. Security

Spontaneous ad hoc networks are formed by mobile nodes that need to communicate during a reduced time slot; these networks have the same problems as the ad hoc networks, but increased because they are temporal networks formed in a given moment by a group of nodes that often users do not know each other; however, they must work together for the proper operation of the network [18]. The use of cryptographic techniques is needed when safe communications must be guaranteed.

Nevertheless, when we talk about the use of cryptography of private key, many of the outlined protocols assume that the nodes know the session key. The same happens when we talk about the use of cryptography of public key (almost all the protocols begin with the assumption that the participants know the public keys of the nodes with those that want to establish the communication). They provide methods to establish a safe and authentic communication channel, assuming that the participants know the node which they are speaking with. The phase of connection establishment and initial exchange of keys, when the nodes do not know each other yet, is a fundamental topic in the environment of the security in spontaneous networks. Security requirements in spontaneous networks are similar to those in traditional networks: confidentiality, integrity, authentication, nonrepudiation, and availability. Both data and routing information must be protected. The characteristics of ad hoc networks make these requirements much more complex: dynamic topology, restricted bandwidth, different capacity links and high error rates, energy and processing capacity limitations, absence of a central server, and often no prior information in the nodes to build the network [1].

These limitations have to be covered by administration mechanisms and by the cooperation among the nodes to maintain service quality, security, and almost automatic discovery and access to the services. This behavior is similar to the human relationships in the society. Everyone must collaborate to maintain a secure world, to improve our quality of life, and have updated news. In this society the trust is very important; we know that the data are correct when they come from a person that we trust.

The required configuration services would be very significant depending on the size of the network, the nature of the participants, and the applications to support it. Confidentiality, integrity, availability, and access control with authentication must be offered without central administration and with energy restrictions. They require key generation, management, and distribution schemes that can be run on small CPUs.

Two fundamental areas must be addressed when we wish to create a spontaneous wireless network security comparable to the traditional networks. First, there must be a trust establishment, key management, and membership control, and, second, there must be network availability and routing security [19].

Our goal is to develop techniques in order to enable the creation of small- and medium-scale ad hoc networks based on the spontaneity of both human interactions and relationships of trust. Given that wireless connectivity is based on physical proximity, it reflects the ways human beings interact. People who are near each other can communicate, exchange things with each other, and ask people to relay information to others. This is all done with an appropriate level of security.

To get an appropriate level of security we establish several protection mechanisms as follows.

(i) *Identification of the Nodes.* It will avoid forging nodes. The proposed solutions are based on the use of cryptography mechanisms. In [20], Stajano and Anderson proposed the use of threshold cryptography through the creation of an authority of distributed certification. The participant $n$ of the network knows a secret and $k$ of them are able to reassemble it. Variations of this outline allow the distribution of the public key and the signatures easily. Another method was proposed in [21] by Zhou and Haas. This method uses a trust web-like PGP where each participant creates a pair of keys, public-private, of its property. When a node is sure of the identity of other node, it signs the correspondent public key certifying its identity. If A certifies the identity of B, the identity of B can be verified.

(ii) *Prevention of Proud Behavior*. In the related literature several solutions were proposed to fight this behavior. One of them is the creation of a virtual currency called nugget [22]. You obtain nuggets by forwarding packets from one node to another, and you spend them when you try to send your own data. You cannot send your packets if you do not have nuggets to pay. The inconvenience is that it should have a trust-specific hardware to assure the currency. Another method is based on the detection and expulsion of a proud node by means of the use of a guardian dog that checks whether the data are transmitted through where it should be. It also intends the use of distributed intrusion detection systems such as IDS, CORE, and CONFIDANT [23]. Another suggestion is the use of MobIDS (Mobile Intrusion Detection System) which is focused on the integration with other mechanisms and with sensors for the detection of proud nodes [24].

(iii) *Security in Routing Protocols against Manipulations.* The main objective of a routing algorithm is to establish an appropriate route between each pair of nodes. If the result of this algorithm is manipulated, the normal operation of the MANET will probably be seriously affected. The means of prevention, such as cypher texts and authentication, will allow defending against some of the attacks. The internal attacks come from compromised nodes belonging to the network. This is a more serious attack, since it is usually more difficult to detect and to be counterattacked. Kargl et al. developed the Secure Dynamic Source Routing (SDSR) in order to prevent these types of attacks [25]. This protocol is a part of SAM (Security Architecture for Mobile Ad Hoc Networks). It provides consistent security in the use of MANET-IDs for the identification of the nodes, SDSR for the routing protocols, and MobIDS for the detection of selfish nodes.

## 6. Protocol Procedure and Messages

Our proposal is based on the use of two information structures: an identity card (IDC) and a certificate. All devices must have an IDC with the following data:

(1) logical identity (LID) that is the logical identity of the user and is unique for each user (it could have its first name and last name, picture, etc.),

(2) public and private keys of the user ($K$) having only the public key sent to other users,

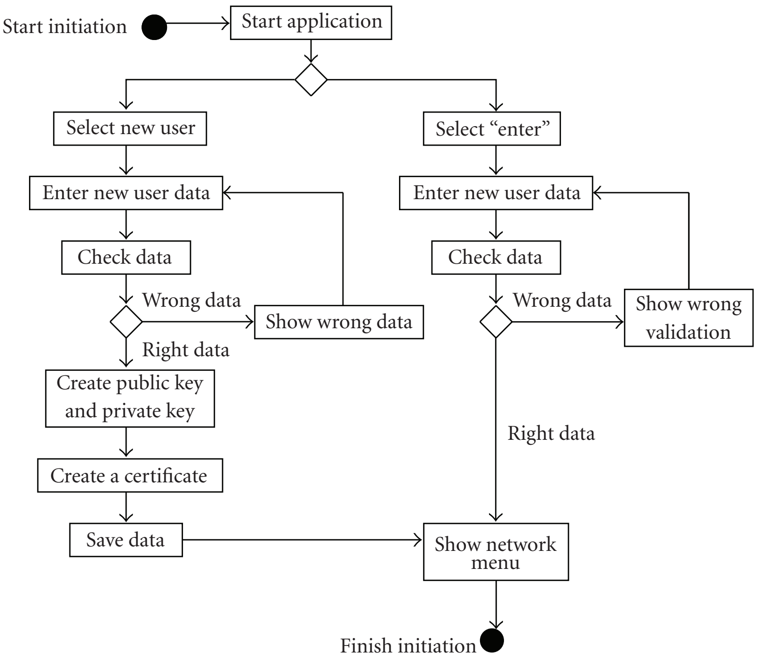

Figure 4: Initiation procedure.

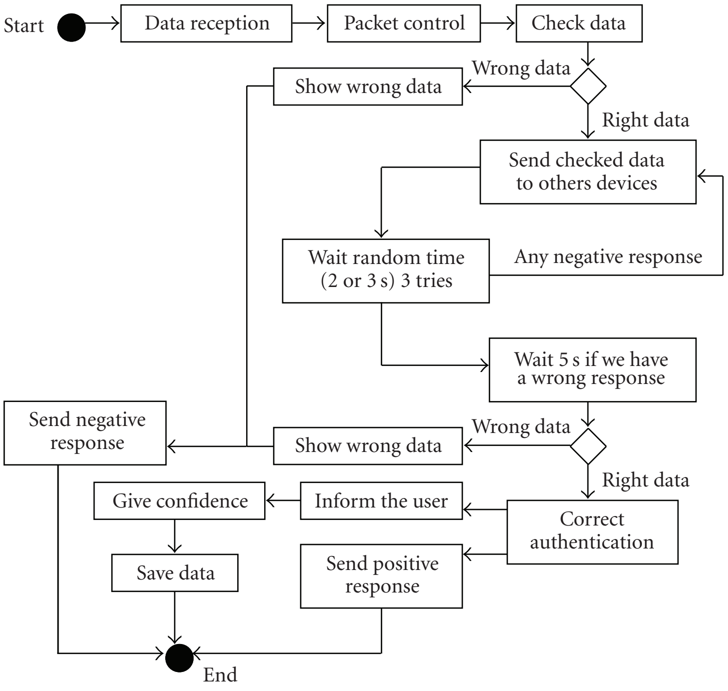

Figure 5: Authentication response procedure.

(3) creation date and expiration date,

(4) IP proposed by the user,

(5) information signature.

The first node in the network generates a network key randomly and waits for any new connection. When a new device wants to join the network, it has to exchange its IDC with the first node, so they start the preauthentication phase. The pre-authentication method has been proposed by several authors as a system to improve the security in ad hoc networks [26, 27]. But in the proposed model, bearing in mind the limitations of the devices, we have to secure the start of the network and the addition of new members, and then, we have to provide integrity (by using hash functions), privacy, and confidentiality to the network. The connection is performed through a short-range technology,

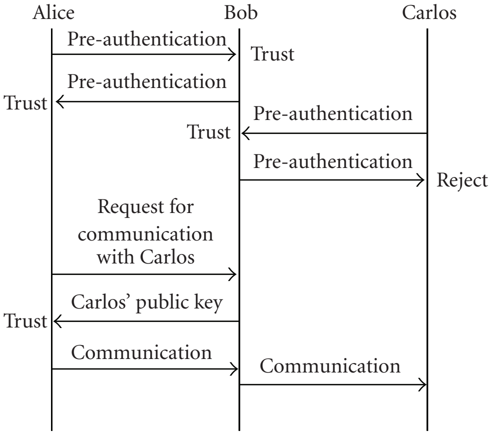

Figure 6: Devices procedure example.

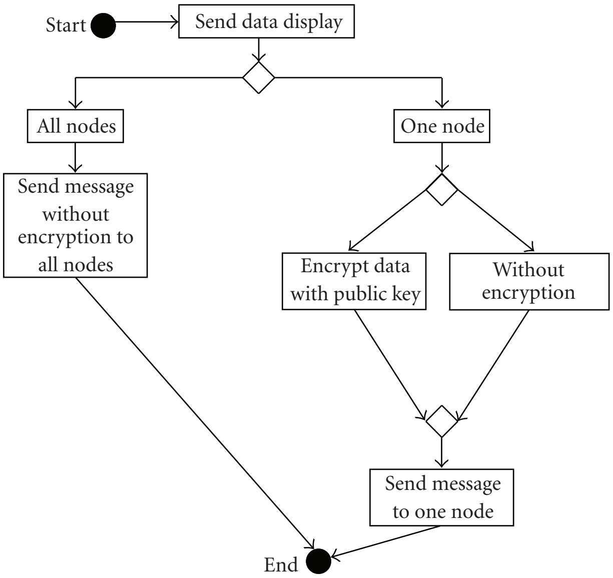

Figure 7: Data transmission procedure.

allowing a face-to-face meeting, so users are known personally (spontaneous networks are given in a limited space). The technology allows flexible data exchange with multiple nodes in the wireless access network. A new node could be authenticated by any of the nodes in the network because they are face to face.

In order to provide secure connections, we use a model based on the use of the asymmetric infrastructure of public keys (higher security than the symmetric cryptography), which is mainly used in the distribution and key management processes. The public key is distributed to the other devices, but the private key is stored confidentially. In our model, because there is not a central Certificate Authority (CA), the authentication is given distributed by the trusted devices. Trusted devices verify the identity of the devices and the validity of the public keys. A device will trust the identity of another device and its public key only if it obtained its certificate through a trusted device of the network.

We used a criterion based on human relationships (social proximity): the trusted networks [28, 29]. When the certificate has been generated, the validity process by the rest of the devices of the network will be based on the trust management process. In this environment, any user can be the certification entity, validating the IDC of the other user. A public key of other user is not considered valid by the other user until a trusted user recognizes that this device is the owner of the key and certify its validity (signing it with its private key). That is, the keys of other devices are valid for a given device only if they have been sent by its trusted devices, otherwise the keys validity cannot be guaranteed. When a user receives a valid certificate, because it is signed by a trusted device, it signs the key with its private key giving authenticity and integrity to the process.

In this process, a user gives trust just to the users that it trusts when obtaining its IDC in the pre-authentication process. Devices act as clients and as servers simultaneously.

The data in the IDC are used to configure the device in case of a successful pre-authentication. The device, which is inside the network, has to check whether the data have been generated correctly and whether there is another device in the network with the same configuration. The device in the network has also to proof that the IDC is being used by its owner and it is not being modified in the transmission process. The message integrity prove is done by checking the received fingerprint. On the other hand, the identity verification can be realized visually by the user when the preauthentication phase is being performed. When the process is finished, the roles are exchanged and the new device can authenticate new devices in the network, even the ones that are in the network. When both devices are pre-authenticated successfully, any information exchanged between them is considered valid. Then, the older device in the network sends to the new one the network key. This key is coded asymmetrically with the public key of the new device and only the new device will be able to decode it with its private key.

The procedure of the application running in the device is as follows. First, the user has to choose between the creation of a new user in the device or two starts as an existing user. The creation of a new user involves the following steps (Figure 4 shows the initiation procedure in detail).

(1) Create the user data (first name, last name, e-mail, password, etc.) and check that they are right.

(2) Create the asymmetric keys (public and private keys).

(3) Generate the user certificate.

When a device joins the network, it sends an authentication message with its data to its neighbor devices. When the neighbor devices receive this message, first it controls the packet and checks the data,to check whether the datahave been altered during the communication. Then, it sends a broadcast message to other devices of the network in order to know that the received data (name, e-mail, and IP) are not being used in the network. This check procedure is performed two times more randomly in order to avoid cross-checking because there could be two devices sending

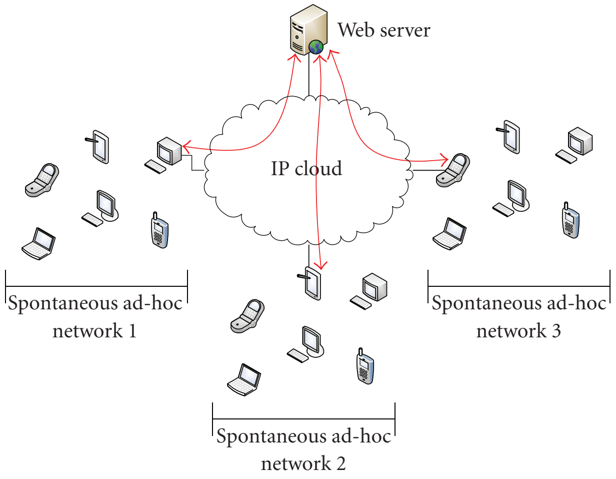
Web server
IP cloud
Spontaneous ad-hoc network 1
Spontaneous ad-hoc network 2
Spontaneous ad-hoc network 3

Figure 8: Simulated network diagram.

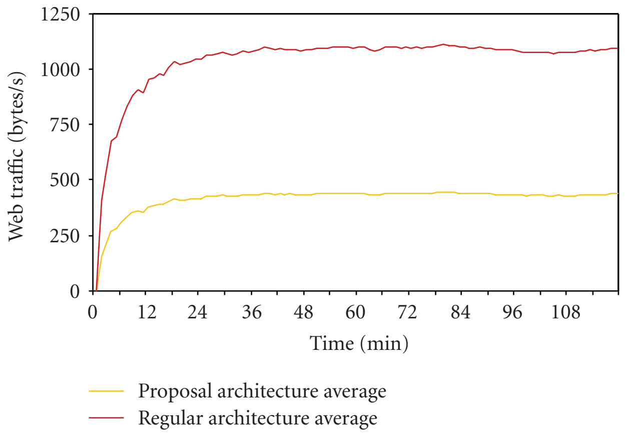

Figure 9: Http traffic through the IP cloud.

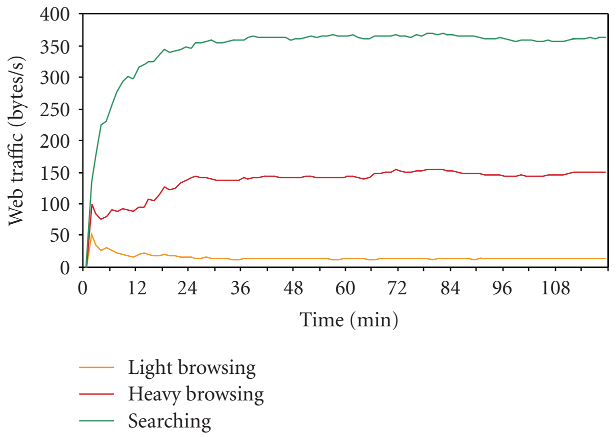

Figure 10: Traffic carried our by gateway nodes.

the same information at the same time. In case of no problem, the user is validated and it is trusted by its neighbor. Then, it stores the data received from its neighbor and sends its public key in case of a positive authentication. These steps are shown in Figure 5.

Let us consider three users: Alice, Bob, and Carlos (see Figure 6). Alice's key is in the set of keys stored by Bob. Bob has signed it to show their agreement. Moreover, Alice is quite demanding when she has to sign keys from other users, so Alice becomes a reliable person for Bob. All keys signed by Alice will be considered valid by Bob, so Alice is a Certificate Authority for Bob. Now, Carlos joins the network and Bob trusts Carlos, but Carlos does not trust Bob (it is not reciprocal). Let us suppose that Alice wants to communicate with Carlos, but she does not have its public key. Then, she asks Bob Carlos' public key. Because Bob trusts Carlos, he sends the public key of Carlos signed by him. So, Alice has Carlos' key and can establish a communication with him to share services. But, in this case, Carlos cannot initiate any communication with Alice because Carlos does not trust Bob, so he has to look for another trusted device to find the public key of Alice. However, we can add to the model the possibility of trust based on time (Carlos could trust Alice after a period of time).

When a device has to send data to another device, it can send the data to a unique destination or to all devices in the spontaneous network. If they are sent in plain text, all the devices will receive the data, but if the data are coded with the private key only the devices with the public key of the source user (those that trust it) will be able to decode the data. These steps are shown in Figure 7.

We have defined two trust levels, but more levels can be added as follows.

(i) *Zero Level*. There is not trust because there has not been any authentication process with this device, it does not trust that device or because the trust level has been put down.

(ii) *First Level*. The device trusts the other device. It has been achieved because an authentication process has been performed.

We have defined two validity levels, but more levels can be added as follows.

(i) *Zero Level*. The key is not valid because it has neither been obtained in the validation process nor a trust device.

(ii) *First Level*. The key is valid because it has been obtained in a pre-authentication process or through a trust device.

The trusted network grows through the Exchange of IDCs between devices. In our proposal the devices do not need to keep all the public keys of the network and the information of all devices inside of it. So, a device does not need to broadcast the authentication information of a new node. The authentication information is distributed through the network.

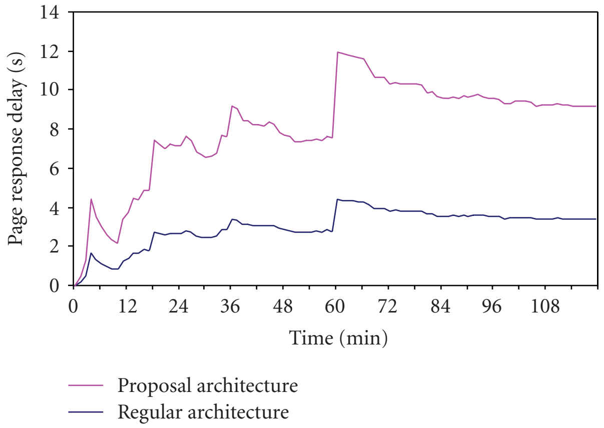

Figure 11: Average delivery delay.

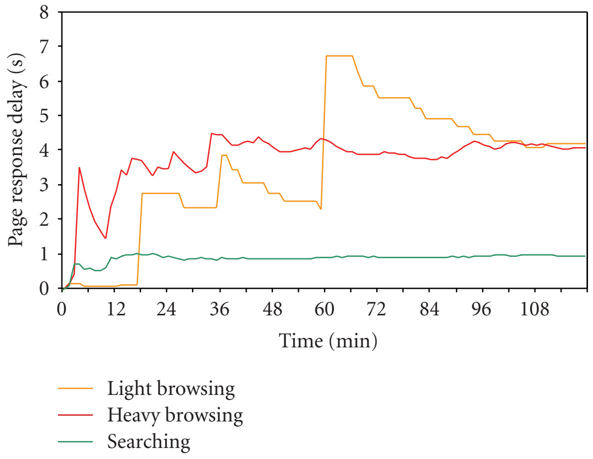

Figure 12: Average Web page delivery delay.

## 7. Performance Analysis: Using Caching Technics

One of the characteristics that distinguish our protocol from others is the caching feature. A lot of authors have researched on caching issues in both wired and wireless networks [30–39]. All of them used a stateful or a stateless server depending on they keep the relation between data and clients which have cached the data or whether they do not do it. On stateful approach the server will be in charge of sending messages of updating each time a data item is modified [35]. If we work with stateless approach, the client will be in charge of sending messages to the server to verify the validity of the cached data before using them [37].

The wide deployment of web cache sharing is currently hindered by the overhead of the Internet Cache Protocol (ICP) [40]. This protocol discovers cache hits in other proxies by having the proxy multicasting a query message to the neighboring caches whenever a cache miss occurs. When the number of proxies increases, both the total communication and the total CPU processing overhead increase quadratically. For this reason, this protocol could not be appropriated when we work on spontaneous networks, due to their limitations.

In [41], a study to demonstrate the benefits of cache sharing, measure the overhead of the existing protocols, and propose a new protocol called “summary cache” is presented. Under this protocol, each proxy keeps a compact summary of the cache directory of every other proxy. When a cache miss occurs, a proxy first probes all the summaries to see whether the request might be a cache hit in other proxies, and sends query messages only to those proxies whose summaries show promising results. In spontaneous networks, each node could work as a proxy; however, these networks are generated to carry out a task that is limited over the time, using services inside the network and with few traffic sent and received from Internet. So, there will not be too much data to be cached.

Dykes presents an analysis of cooperative proxy caching in [42]. Sometimes it is not clear the viability of this type of caching because a remote proxy could not be inherently faster than the origin Web Server. However they show how proxy cooperation can potentially reduce the variability in response time, the number of long delays, and congestion of busy Web servers. Their analysis examines the interaction of three discovery mechanisms with the mesh and hierarchy: ideal discovery, query-based discovery such as the Internet Cache Protocol [33–44], and directory-based discovery using cache digest exchanges. In query methods, clients locate cached copies by sending queries to member of the cache group. In directory methods, propagation of HTTP metadata can, however, affect both user response times and hit ratios. They conclude that cooperative proxy caching is marginally viable for a mesh organization, but it is not viable for a general hierarchical design.

In [45], a semantic caching scheme is used to access location-dependent data in Mobile computing. They developed the semantic cache replacement strategy called FAR, which aims to let the cache contents move as the user moves. The problematic of updating in caching systems is tackled in [46]. They propose a pull-based approach, called aggregate cache based on demand (ACOD) scheme that uses an efficient search algorithm for finding the queried data items.

Other authors, as Park et al. in [47], discuss another problematic issue such as the gateway discovery. They propose a load-adaptative access gateway discovery protocol and a QoS-enabled access gateway selection scheme that can exploit relevant network conditions. The gateway is selected based on the number of hops and the capability of them. This protocol enables lower average delay compared to the others and less overhead compared to the proactive and the hybrid approach because it dynamically adjusts access gateways’ proactive area based on the offered load. However, the problematic of distributed data caching is not studied.

There are more proposals presented in [48–52]. Some of these methods propose, for example, an algorithm of two nodes [32], which share the data to avoid duplications or a replicated data. It allows improving the data accessibility [35].

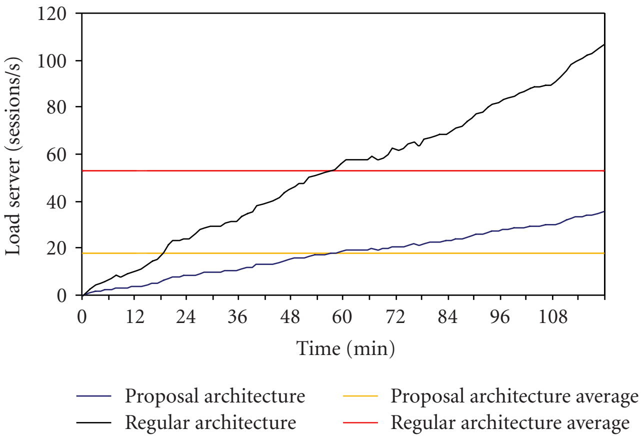

Figure 13: Load on the Web server.

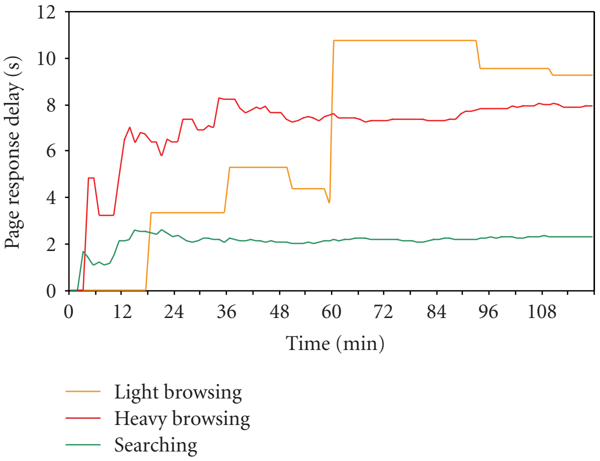

Figure 15: Average web page delivery delay on the Web server.

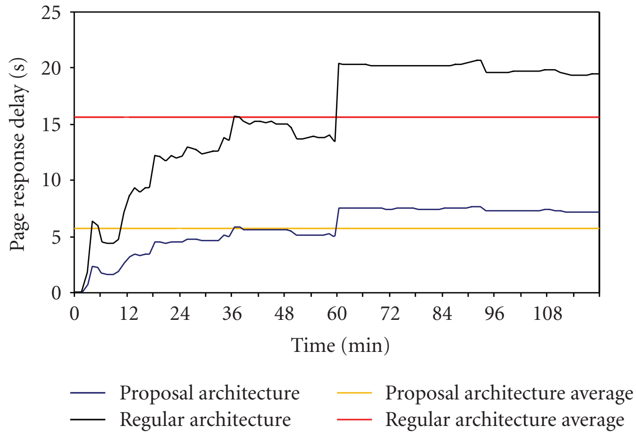

Figure 14: Request and delivery delays.

In [36], a mechanism to update propagation mechanism on peer-to-peer networks (P2P) is proposed. In this model, intermediate client caches the index entries to locate the node where the contents are cached or stored in order to reduce the access latency and to balance the workload. Intermediate clients' index entries are maintained by propagating the updated index entries. However, cached index entries may become obsolete due to mobility of MTs (mobile terminals) that incurs changes in the network topology in mobile environments. In [53], a system for stateless server is proposed. Servers periodically broadcast an invalidation report (IR), in which the updated data items are marked. On the other hand, in order to reduce the latency, a new type of packet is introduced between IR packets, the updated invalidation report (UIR).

In MANETs and Spontaneous networks, we should take into account that the nodes inside them could have several limitations such as limited battery or stored capacity, insufficient wireless bandwidth, or limited accessibility to the wired Internet. It could be several limitations such as longer message latency. In MANETs, nodes could be limited in their resources. To save wireless bandwidth and reduce the average latency will be very important. For this reason, caching will be a very important topic to be taken into account.

There are several proposals for MANETS presented in [53–58]. Some of them are based on replication schemes and periodical updating using effective replica allocation [53]. Others are based on dissemination strategies [58], where a subset of the terminal mobiles are selected as servers and where the updating/queering operations are carried through several proposed heuristics. A cooperative caching scheme for IMANET (Internet-based mobile ad hoc network [59]) environment is proposed in [56]. It lets to increase data accessibility by peer-to-peer communication among the mobiles terminals. In [60], the authors propose, on the same environment (IMANET), an aggregate caching mechanism. In these networks, usually, the mobile terminals have direct access to the Internet, although some terminals could not have direct access or could be disconnected from it due to the mobility. In these cases, nodes could access the requested data items from the local cache or nearby mobile terminals or via relays. The main problem in this proposal is the necessity of both sending broadcasts request to search to locate the requested data and the process of cache data managing. In our proposal, the node in charge of proving access to WWW (the Access Point node) is the one that manages the caching data. Depending on the resources, the AP node establishes the resources spent on this process. Usually the node with more resources and internet access will be chosen to perform this task. A 7Ds architecture is proposed in [55]. In their architecture, several protocols share and disseminate information among users. Their proposal can operate either in a prefetch mode or in a demand mode.

Spontaneous networks are built to form temporarily networks, with a small dependence or without a central administration, and without expert users' intervention, to solve a problem or to develop a certain task. All aspects of auto-configuration and management have to be carried out by the nodes that form the network.

We propose two proposals. In our first proposal only one node (the ones selected to provide the access to Internet) caches the information. It avoids the multicast messages needed to know where the cached data are (when we work on distributed system). In the advanced proposal, the system works with a proactive system where nodes obtain data on a distributed way. In this proposal, the nodes, which work together to carry out an established task, report to the other nodes about their cached domains, which let to avoid the updating multicast traffic. Each node will be able to access to different domains and inform about both their services and the cached portals to the other nodes (it will be in charge of keeping updated the data about several services and domain). If the node in charge will not have the data, it will ask for them to Internet. Since we work on this second proposal, only with domains, the traffic is less than the necessary to maintain all http metadata updated. Since our traffic is low and each node decides of which domain will be cached, the metadata do not change so much and it will not be necessary to send too much updated messages. When the number of accessed portals increases, the AP node will inform to the other nodes. Our system lets to authenticate the terminal mobiles and autoconfigure the networks.

In Table 1 we can see the comparative of several different caching schemas and both proposed systems.

## 8. Protocol Validation

*8.1. Test Bench.* In order to evaluate our proposal we simulated the diagram shown in Figure 8 with the OPNET Modeler simulator [61]. We created a situation where there is a Web server connected to an IP cloud, which simulates Internet behavior. Three spontaneous networks are connected to this IP cloud. Each one performs different type of http traffic in order to test how its performance is. The type of traffic used is as follows.

(1) *First Spontaneous Network: Light Browsing*. This http traffic used HTTP 1.1. The average time between entries (page requests) followed an exponential distribution of an average of 720 seconds. The page had a constant size of 500 bytes in text and 5 small images of 50 bytes each. This traffic was served following an exponential distribution of an average of 10 pages. The type of service was Best Effort.

(2) *Second Spontaneous Network: Heavy Browsing*. This http traffic used HTTP 1.1. The average time between entries (page requests) followed an exponential distribution of an average of 60 seconds. The page had a constant size of 1000 bytes in text and 5 medium-sized images of 100 bytes each. This traffic was served following an exponential distribution of an average of 10 pages. The type of service was Best Effort.

(3) *Third Spontaneous Network: Searching*. This http traffic used HTTP 1.1. The average time between entries (page requests) followed an exponential distribution of an average of 10 seconds. The page had a constant size of 1000 bytes in text and 2 medium-sized images of 100 bytes each. This traffic was served following an exponential distribution of an average of 2 pages. The type of service was Best Effort.

We have simulated two scenarios. In the first scenario we simulate our proposal. When there is an http request to the web server, only the node that provides WWW access replies. In the second scenario, it is simulated a regular Ad hoc architecture, where each http request is sent to the server and it replies with the adequate content. Both scenarios have been simulated for three times and then we have selected the most representative simulation.

Each spontaneous ad hoc network is made up of five or six devices. The nodes in the topology have the following features: 40 MHz processor, a 512 KB memory card, a radio channel of 11 Mbps, and 2.4 GHz as the working frequency. We selected AODV as the routing protocol for the spontaneous networks, but it could be changed. The node carrying out the gateway task between the wired IP cloud and the spontaneous ad hoc network has the same characteristics as the other nodes, but an Ethernet interface is added.

We forced failures in the spontaneous networks with the recovery processes. This allowed us to observe the network behavior when there are physical topology changes and node failures. Failures and recoveries usually happen in these kinds of networks, so we wished to study how a network-level protocol works when those events occur. Those errors always take place in the weak nodes (the most stable ones are the gateway node and the web server).

The IP cloud allows the modification of some parameters in order to give a more realistic feel to our simulation. In our case we varied the delay of the packets by introducing 2 seconds and also a rejected packet rate of 1%. Finally, we emphasize that each spontaneous network carried out some sort of http request to the server or node connected to Internet.

*8.2. Simulations Measurements.* Figure 9 shows the http traffic going through in the IP cloud. We observe the behavior of our architecture and we compare it with the regular architecture. Note that the average http traffic in our proposal is around 430 bytes. It can be compared to the regular architecture which has an average of 1090 bytes, therefore displaying a 61% of improvement. In an instantaneous traffic analysis, we observe that the traffic is more stable and displays fewer fluctuations in our architecture.

As it has been indicated in the test bench, in each spontaneous network we have simulated different type of web consultation. In one network we have typical Web traffic of “Light Browsing”, in a second one “heavy browsing”, and finally, in the third one, the “searching” type. In Figure 10 we observe the load on the gateway node connected to Internet. All the Web consultations finish at this node. We see that the Web traffic “light browsing” bears little load; only sporadic consultations are carried out. In the case of “heavy browsing” consultations, we see that there is higher http load traffic with peaks at specific time intervals. Finally, the highest load traffic is the “searching” type. It is because a lot of consultations are made in a short time period.

In Figure 11 we observe the average delivery delay of the Web pages. It can be seen that, once the networks converge, the average delivery delay is around 9 seconds in the regular

Table 1: Comparison between cache techniques.

| | Lim et al. [46] | Dykes and Robbins [42] | Ren and Dunhan [45] | Fan et al. [41] | Proposal 1 | Proposal 2 |
|---|---|---|---|---|---|---|
| Auto-configuration of the network | 0 | 0 | 0 | 0 | 3 | 3 |
| Auto-configuration of the nodes | 0 | 0 | 0 | 0 | 3 | 3 |
| Use of central servers or proxies | 1 | 3 | 3 | 3 | 0 | 0 |
| Distributed network caching | 3 | 3 | 3 | 3 | 0 | 3 |
| Internet access provided by any node (one or several selected by the nodes of the network) | 3 | 0 | 0 | 0 | 3 | 3 |
| Internet access provided by one node (fixed and not dynamic server) | 0 | 0 | 0 | 0 | 0 | 0 |
| Cached data by the network's nodes | 3 | 0 | 3 | 0 | 0 | 3 |
| Scalability | 2 | 3 | 3 | 3 | 3 | 3 |
| Cached data by the network's AP/proxies | 0 | 3 | 0 | 3 | 3 | 3 |
| Flexibility | 2 | 0 | 3 | 3 | 3 | 3 |
| Intranet broadcasting necessary (searching data/updating metadata) | 3 | 3 | 0 | 3 | 1 | 1 |

*Notation:* (0) Not contemplated, (1) Not necessary, (2) Supposed (not explained but contemplated), (3) Contemplated (taking into account and explained).

architecture and 3.5 seconds in the proposed spontaneous architecture. There is an improvement of 62%. We can also observe in the fluctuations of each one of the curves that our architecture is more stable.

In Figure 12 we can see the average Web page delivery delay from nodes on our ad hoc spontaneous networks. In this graph we observe that the traffic with the shortest delay is the "searching" type. This is given because, when we do a search, the browsers provide information which does not consume much bandwidth and so the delay is shorter. With respect to the other two types of traffic, we can see that, once the network converges, then the average delay is around 4 seconds. In these cases the delay is longer because this type of traffic is heavier and therefore there is needed more delivery time to download the full web page. In the light browsing traffic we can see that there are 2 times (instant 18 and instant 60), where the delay is quite high. This is due to the type of simulated traffic. This type of traffic is burst traffic, where the probability that an object has been previously requested site is low. The petitions must bear the main server.

Figure 13 represents the load on the Web server using each one of the proposed traffic. In our proposed architecture, the central web server has an average load of 17.5 sessions/s. However this load increases 67% (52.6 sessions/s) when we have the same web requests on a regular architecture. Our architecture performs lower load on the server. This is owing to the structure of the architecture itself. In the regular architecture, each request is made to the server, while in our proposal, if the connected node has already carried out the consultation, this request is stopped in the gateway node which then takes the task of sending the web page, taking a role similar to cache servers. This means less traffic for the IP cloud and a shorter delay; however information which is not updated may be at a given time.

In Figure 14, it is shown the average delay from the Web server to a user device until all Web information is received. We observe that in the regular architecture, once the network has converged, this delay is around 15.6 seconds. A relatively long delay is mainly due to the ad hoc networks and the response time of the Web server. In contrast, this is improved by about 66% (around 5.7 seconds) when our proposal is used. This improvement is given because of the proximity of the Web resources. The Web pages are found in the gateway node connected to Internet.

Figure 15 shows the delay average web page response on the Web server. It can be observed that the "searching" traffic is almost constant during all the time. Its mean delay is about 2.12 seconds approximately. This is because this traffic profile has less data exchange. The "light browsing" traffic profile is not stable as it has several peaks. We see that from 18 minutes to 60 minutes the average page delay is about 4.4 seconds. But this delay increases up to 10.7 seconds during the following 36 minutes. This can occur because the time between arrivals used in our simulations is quite high. Finally, the "heavy browsing" traffic has an increasing behaviour in the first stage (the first 18 minutes), until it got stable around a value of

7.62 seconds. This mean value is higher than the "searching" profile because in this case all the information is sent to the final user.

## 9. Conclusion

In this paper we propose a secure spontaneous ad hoc protocol that allows groups of users to collaborate in a given time to accomplish a collaborative task. The main differences with regular ad hoc networks are that these networks are created to develop one task on a limited period of time and on a specific space. The communities then allow the exchange of knowledge and teamwork in order to carry out a task and the collaboration or exchange of information between various communities to complete work. The resources are provided by the different members of the community, permitting the access of all members. A quick and easy auto-configuration of both networks, and the establishment of the security are two fundamental issues. The proposed solution presents a distributed model, where the interaction required between devices is minimal. It allows the access to different services offered through different nodes of the network. In this model all the nodes collaborate for the proper operation and management of the network. The devices do not need to keep all the security data of the network and the information of all devices inside of it. The association between nodes is set up when they are close to each other, that is, in presentation and greeting. The protocol allows the communication with other spontaneous ad hoc network through Internet. It can be performed by giving a certificate generated by a recognized Certification Authority (it will have clearance to work through the WWW) to the gateway node, which provides access to Internet. This digital certificate acts as a unique identifier within the network and will allow the possessor to be identified in Internet. It will be needed to guarantee both the authenticity of the communities and the integrity of the transmitted information.

In addition, the limited resources of nodes have been taken into account, establishing caching techniques that let reduce the overload of the nodes. We have validated the success of our proposal simulating what happens when all http information requested by the ad hoc network has to be performed through the gateway node and when there is a spontaneous network sharing the resources of the devices.

The proposal has been analyzed analytically. We have shown the device procedure and the protocol messages designed for its proper operation. Finally, we have compared our proposal with other cache techniques published in the literature. Although spontaneous ad hoc networks are well referenced in the literature, there are very few deployments until today. Moreover, they can be developed for any type of infrastructure [62].

## Acknowledgments

Authors want to give thanks to the anonymous reviewers for their valuable suggestions, useful comments, and proofreading of this paper. This work was partially supported by the Ministerio de Educación y Ciencia, Spain, under Grant no. TIN2008-06441-C02-01, and by the "Ayudas complementarias para proyectos de I+D para grupos de calidad de la Generalitat Valenciana" (ACOMP/2010/005).

## References

[1] S. Preuß and C. H. Cap, "Overview of spontaneous networking-evolving concepts and technologies," in *Rostocker Informatik-Berichte*, vol. 24, pp. 113–123, Fachbereich Informatik der Universit at Rostock, 2000.

[2] S. Gallo, L. Galluccio, G. Morabito, and S. Palazzo, "Rapid and energy efficient neighbor discovery for spontaneous networks," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, October 2004.

[3] 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.

[4] J. Latvakoski, D. Pakkala, and P. Pääkkönen, "A communication architecture for spontaneous systems," *IEEE Wireless Communications*, vol. 11, no. 3, pp. 36–42, 2004.

[5] V. H. Zarate Silva, E. I. De Cruz Salgado, and F. R. Quintana, "AWISPA: an awareness framework for collaborative spontaneous networks," in *Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference (FIE '06)*, pp. 1–6, October 2006.

[6] 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.

[7] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, August 1994.

[8] D. B. Johnson, D. A. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, Ad Hoc Networking*, Addison-Wesley Longman Publishing, Boston, Mass, USA, 2001.

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003.

[10] V. Park and M.S. Corson, "IETF MANET Internet Draft "draft-ietf-MANET-tora-spe03.txt"," Novemmer 2000.

[11] A. C. Viana, M. D. De Amorim, S. Fdida, and J. F. de Rezende, "Self-organization in spontaneous networks: the approach of DHT-based routing protocols," *Ad Hoc Networks*, vol. 3, no. 5, pp. 589–606, 2005.

[12] R. L. Gilaberte and L. P. Herrero, "IP addresses configuration in spontaneous networks," in *Proceedings of the 9th WSEAS International Conference on Computers*, Athens, Greece, July 2005.

[13] A. C. Viana, M. Dias de Amorim, S. Fdida, and J. F. de Rezende, "Self-organization in spontaneous networks: the approach of DHT-based routing protocols," *Ad Hoc Networks*, vol. 3, no. 5, pp. 589–606, 2005.

[14] J. I. Alvarez-Hamelin, A. Carneiro Viana, and M. Dias De Amorim, "Architectural considerations for a self-configuring routing scheme for spontaneous networks," Tech. Rep. 1, October 2005.

[15] R. Lacuesta and L. Peñalver, "Automatic configuration of ad-hoc networks: establishing unique IP link-local addresses,"

in *Proceedings of the International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '07)*, Valencia, Spain, October 2007.

[16] E. F. Foulks, "Social network therapies and society: an overview," *Contemporary Family Therapy*, vol. 3, no. 4, pp. 316–320, 1985.

[17] Y. Wang and H. Wu, "DFT-MSN: the delay/fault-tolerant mobile sensor network for pervasive information gathering," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[18] T. Kindberg and K. Zhang, "Validating and securing spontaneous associations between wireless devices," in *Proceedings of the 6th Information Security Conference (ISC '03)*, pp. 44–53, Springer, 2003.

[19] J. Al-Jaroodi, "Routing security in open/dynamic mobile ad hoc networks," *The International Arab Journal of Information Technology*, vol. 4, no. 1, pp. 17–25, 2007.

[20] F. Stajano and R. J. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks," in *Proceedings of the 7th International Workshop on Security Protocols*, pp. 172–194, April 1999.

[21] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[22] M. Hauspie and I. Simplot-Ryl, "Cooperation in ad hoc networks: enhancing the virtual currency based models," in *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense '06)*, Nice, France, May 2006.

[23] X. Wang, F. Dai, L. Qian, and H. Dong, "A way to solve the threat of selfish and malicious nodes for ad hoc networks," in *Proceedings of the International Symposium on Information Science and Engieering (ISISE '08)*, vol. 1, pp. 368–370, Shanghai, China, December 2008.

[24] F. Kargl, A. Klenk, M. Weber, and S. Schlott, "Sensors for detection of misbehaving nodes in MANETs," in *Detection of Intrusion and Malware and Vulnerability Assessment (DIMVA '04)*, pp. 83–97, Dortmund, Germany, July 2004.

[25] F. Kargl, A. Geiss, S. Scholott, and M. Weber, "Secure dynamic source routing," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, Big Island, Hawaii, USA, January 2005.

[26] S. Gokhale and P. Dasgupta, "Distributed authentication for peer-to-peer networks," in *Proceedings of the Symposium on Applications and the Internet Workshops*, pp. 347–353, January 2003.

[27] S. Capkun, L. Buttyán, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, 2003.

[28] F. Stajano and R. Anderson, "The resurrecting duckling security issues for ad-hoc wireless networks," in *Proceedings of the 7th International Workshop on Security Protocols*, vol. 1796 of *Lecture Notes in Computer Science*, pp. 172–194, Springer, Berlin, Germany, 1999.

[29] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: authentication in ad-hoc wireless networks," in *Proceedings of the International Symposium on Network and Distributed Systems Security (NDSS '02)*, San Diego, Calif, USA, February 2002.

[30] D. Barbara and T. Imielinski, "Sleepers and workaholics: caching strategies in mobile environments," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1–12, May 1994.

[31] G. Cao, "A scalable low-latency cache invalidation strategy for mobile environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1251–1265, 2003.

[32] Q. Hu and D. Lee, "Cache algorithms based on adaptive invalidation reports for mobile environments," *Cluster Computing*, vol. 1, no. 1, pp. 39–50, 1998.

[33] J. Jing, A. Elmagarmid, A. Helal, and R. Alonso, "Bit-sequences: an adaptive cache invalidation method in mobile client/server environments," *Mobile Networks and Applications*, vol. 2, no. 2, pp. 115–127, 1997.

[34] A. Kahol, S. Khurana, S. Gupta, and P. Srimani, "An efficient cache management scheme for mobile environment," in *Proceedings of the 20th International Conference on Distributied Computing System (ICDCS '00)*, pp. 530–537, Taipei, Taiwan, April 2000.

[35] M. Kazar, "Synchronization and caching issues in the Andrew file system," in *Proceedings of USENIX Conference*, pp. 27–36, Dallas, Tex, USA, February 1988.

[36] M. Roussopoulos and M. Baker, "CUP: controlled update propagation in peer-to-peer networks," in *Proceedings of USENIX Annual Technical Conference*, San Antonio, Tex, USA, June 2003.

[37] S. Sandberg, S. Kleiman, D. Goldberg, D. Walsh, and B. Lyon, "Design and implementation of the sun network file system," in *Proceedings of USENIX Summer Conference*, pp. 119–130, Portland, Ore, USA, June 1985.

[38] K. Wu, P. S. Yu, and M. Chen, "Energy-efficient caching for wireless mobile computing," in *Proceedings of the 12th IEEE International Conference on Data Engineering*, pp. 336–343, New Orleans, La, USA, February-March 1996.

[39] M. K. H. Yeung and Y.-K. Kwok, "Wireless cache invalidation schemes with link adaptation and downlink traffic," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 68–83, 2005.

[40] D. Wessels and K. Claffy, "Internet cache protocol (IC) v.2," http://www.ietf.org/rfc/rfc2186.txt.

[41] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 281–293, 2000.

[42] S. G. Dykes and K. A. Robbins, "A viability analysis of cooperative proxy caching," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1205–1214, Anchorage, Alaska, USA, April 2001.

[43] D. Wessels and K. Claffy, "RFC 2186: Internet cache protocol (ICP), version 2," The Internet Engineering Taskforce, September 1997.

[44] D. Wessels and K. Claffy, "RFC 2187: application of internet cache protocol (ICP), version 2," The Internet Engineering Taskforce, September 1997.

[45] Q. Ren and M. H. Dunhan, "Using semantic caching to manage location dependent data in mobile computing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 210–221, Boston, Mass, USA, August 2000.

[46] S. Lim, W.-C. Lee, G. Cao, and C. R. Das, "Cache invalidation strategies for internet-based mobile ad hoc networks," *Computer Communications*, vol. 30, no. 8, pp. 1854–1869, 2007.

[47] B.-N. Park, W. Lee, and C. Lee, "QoS-aware internet access schemes for wireless mobile ad hoc networks," *Computer Communications*, vol. 30, no. 2, pp. 369–384, 2007.

[48] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications*

*Societies (INFOCOM '01)*, pp. 1568–1576, Anchorage, Alaska, USA, April 2001.

[49] M. Papadopouli and H. Schulzrinne, "Effects of power conservation, wireless converage and cooperation on data dissemination among mobile devices," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 117–127, Long Beach, Calif, USA, October 2001.

[50] P. Can and S. Irani, "Cost-aware WWW proxy caching algorithms," in *Proceedings of the USENIX Symposium on lnternet Technology and Systems*, December 1997.

[51] L. Rizzo and L. Vicisano, "Replacement policies for a proxy cache," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 158–170, 2000.

[52] S. Williams, M. Abrams, C. R. Strandridge, G. Abdulla, and E. A. Fox, "Removal policies in network caches for world-wide web documents," in *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 293–305, Palo Alto, Calif, USA, August 1996.

[53] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 1568–1576, Anchorage, Alaska, USA, April 2001.

[54] T. Hara, "Replica allocation in ad hoc networks with period data update," in *Proceedings of the 3rd International Conference on Mobile Data Management (MDM '02)*, pp. 79–86, Edmonton, Canada, July 2002.

[55] M. Papadopouli and H. Schulzrinne, "Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 117–127, Long Beach, Calif, USA, October 2001.

[56] F. Sailhan and V. Issarny, "Cooperative caching in ad hoc networks," in *Proceedings of the 4th International Conference on Mobile Data Management (MDM '03)*, vol. 2574 of *Lecture Notes in Computer Science*, pp. 13–28, Melbourne, Australia, January 2003.

[57] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.

[58] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information dissemination in partitionable mobile ad hoc networks," in *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS '99)*, pp. 4–13, Lausanne, Switzerland, October 1999.

[59] M. S. Corson, J. P. Macker, and G. H. Cirincione, "Internet-based mobile ad hoc networking," *IEEE Internet Computing*, vol. 3, no. 4, pp. 63–70, 1999.

[60] S. Lim, W.-C. Lee, G. Cao, and C. R. Das, "A novel caching scheme for improving internet-based mobile ad hoc networks performance," *Ad Hoc Networks*, vol. 4, no. 2, pp. 225–239, 2006.

[61] Opnet Modeler, http://www.opnet.com/solutions/network_rd/modeler_wireless.html.

[62] R. Lacuesta, J. Lloret, M. Garcia, and L. Peñalver, "Two secure and energy-saving spontaneous ad-hoc protocol for wireless mesh client networks," *Journal of Network and Computer Applications*. In press.