Lloret, J.; Sendra, S.; Jimenez, JM.; Parra-Boronat, L. (2016). Providing security and fault tolerance in P2P connections between clouds for mHealth services. Peer-to-Peer Networking and Applications. 9(5):876-893. doi:10.1007/s12083-015-0378-3

Providing Security and Fault Tolerance in P2P connections between Clouds for mHealth Services

Article *in* Peer-to-Peer Networking and Applications · June 2015

DOI: 10.1007/s12083-015-0378-3

4 authors:

Jose M. Jimenez
Universitat Politècnica de València
49 PUBLICATIONS **104** CITATIONS
SEE PROFILE

Jaime Lloret
Universitat Politècnica de València
463 PUBLICATIONS **3,278** CITATIONS
SEE PROFILE

Sandra Sendra
University of Granada
102 PUBLICATIONS **872** CITATIONS
SEE PROFILE

Lorena Parra
Universitat Politècnica de València
23 PUBLICATIONS **74** CITATIONS
SEE PROFILE

Providing Security and Fault Tolerance in P2P connections between Clouds for mHealth Services

Jaime Lloret[1], Sandra Sendra[2], Jose Miguel Jimenez[3], Lorena Parra[4]
Instituto de Investigación para la Gestion integrada de zonas Costeras. Universidad Politécnica de Valencia (Spain)
[1]jlloret@dcom.upv.es, [2]sansenco@posgrado.upv.es; [3]jojiher@dcom.upv.es; [4]loparbo@doctor.upv.es

Abstract: The mobile health (mHealth) and electronic health (eHealth) systems are useful to maintain a correct administration of health information and services. However, it is mandatory to ensure a secure data transmission and in case of a node failure, the system should not fall down. This fact is important because several vital systems could depend on this infrastructure. On the other hand, a cloud does not have infinite computational and storage resources in its infrastructure or would not provide all type of services. For this reason, it is important to establish an interrelation between clouds using communication protocols in order to provide scalability, efficiency, higher service availability and flexibility which allow the use of services, computing and storage resources of other clouds. In this paper, we propose the architecture and its secure protocol that allows exchanging information, data, services, computing and storage resources between all interconnected mHealth clouds. The system is based on a hierarchic architecture of two layers composed by nodes with different roles. The routing algorithm used to establish the connectivity between the nodes is the shortest path first (SPF), but it can be easily changed by any other one. Our architecture is highly scalable and allows adding new nodes and mHealth clouds easily, while it tries to maintain the load of the cloud balanced. Our protocol design includes node discovery, authentication and fault tolerance. We show the protocol operation and the secure system design. Finally we provide the performance results in a controlled test bench.

Keywords – Communication between clouds, Fault Tolerance, Protocol, Architecture, eHealth, mHealth, security between clouds.

1. Introduction

Electronic health (eHealth) is a new area of research, which is based on the use of information and communication technologies with health purposes. Several authors propose different definitions for eHealth but one of the best ones is written by the World Health Organization (WHO) [1]: *"eHealth is the cost-effective and secure use of information and communications technologies in support of health and health-related field, including health care services, health surveillance, health literature, and health education, knowledge and research"*. The areas of eHealth include treating patients, conducting research, educating the health workforce, tracking diseases and monitoring public health [2].

The concept of eHealth is not so new, but its real use and development started few years ago. It is expected a huge grow of eHealth in few years due to the increase on chronic diseases prevalence, the ageing of people and the increase of population. These issues are obsoleting the traditional health systems and making them not sustainable.

The services offered under the eHealth topic are large. eHealth enclose from a simple telephone call to the admission area of a hospital to order an appointment, to the most advanced sensors to measure a disease [3]. The commonest services are Electronic Medical Records, Electronic prescriptions, and Patient monitoring. The main benefits of the use of eHealth are shown below:

- Reduction on appointments to the General Practitioner (GP).
- Reduction on the number of hospital admissions.
- Reduction on the rate of sanitary personnel per patients.

- Better patient monitoring.
- Empowerment of patients.
- Less error in prescriptions.
- Decrease costs of sanitary industry.

Some published studies demonstrate the benefits of the use of eHealth technologies. The British Health Department is using eHealth in a specific program that involves 6.191 patients and 238 GP [4]. Moreover, there is a report from the European Union that enhances this information [5]. The highlights of both publications are shown below:

- Mortality rates reduction in 45% [4].
- Emergency hospital admissions reduction in 20% [4].
- Avoid 5 million yearly errors in prescription using Electronic Transfer of Prescriptions [5].
- Avoid 100,000 yearly adverse medication events using Computerised Physician Order Entry and Clinical Decision Support [5].

mHealth is one specific part of eHealth. It can be defined as the use of devices which are wearable and dotted with wireless capability. They have to be able to transmit, store, process and save data between different users (such as patients, pharmacists or doctors) in real or non-real time [6]. The most used devices are smartphones, personal digital assistants, personal monitoring devices and other wireless devices. mHealth services generally offer medication adherence, vital signs' monitoring, activity monitoring and alert systems, wellness and rehabilitation, remote consultation, solutions for caregivers and even for weigh control [7-9]. The wireless technologies used in mHealth are: Wireless Local Area Networks (WLAN), Global System for Mobile communications (GSM), General Packet Radio Service (GPRS), third generation of mobile telecommunications technology (3G) and Satellite Communications [10]. The most important challenges for mHealth are the security, scalability, availability (anywhere/anytime), management of distributed/decentralized/shared space and multilayer management infrastructure [6].

Even if mobile devices are very useful in monitoring and tracking patients due to its small size and portable characteristics, other characteristics are no suitable for mHealth (such as their low memory, computing capacity and weak security [11]). Those characteristics make that most of mHealth systems need other technology to develop competent functions. The technology used in most cases is cloud computing [11]. Cloud can turn the actual healthcare industries into a more sustainable system, by reducing the costs and bringing innovative solutions [12]. The devices can send and store the information in the cloud, while the GP, caregivers, pharmacists, familiars or the patient can access to all the information anywhere and anytime. Different works are developed in mHealth where the service offered uses the cloud computing to store the data of the wearable devices. Some examples are shown above:

- Multimedia data of digital signal processor storage [11].
- Collection and dissemination of health data in real time - electrocardiography (ECG) [13].
- Diagnose and treatment by ECG data collection [14].
- Store data from patient monitoring [15] [16].

Cloud computing is a recent concept based on sharing resources to achieve coherence and economies of scale, similar to a utility over a network. Cloud computing implies deploying groups of remote servers and systems that allow centralized data storage and online access to these services and resources. We can find different kind of networks using cloud computing such as Mobile Ad-hoc Network (MANET) [17] and Vehicular Ad Hoc Networks (VANETs) [18][19], among others. Moreover, we may use them to provide services such as multimedia content [20], file sharing, sensor values, etc.

Cloud-based networks are characterized by their capacity of being easily scalable which tries to improve the network performance, optimize the use of available resources in the network and improve their operation and maintenance. One of the most important problems in the use of cloud computing is its low security. For this reason, one of the main challenges is to provide higher levels of security to those cloud-based networks [21] [22] mainly when communicating the clouds [23] as well as when clients access the infrastructure [24].

According to [12] small clouds are safer that big clouds. The security of health data is a crucial theme when eHealth systems are going to be developed. The medical records data are sensitive information and should be accessed only by doctors, and it must be protected. Several works are performed with this purpose [25-27]. According to [28] almost 30% of the patients prefer that doctors only access to their information if they give their

consent. So it is very important to establish and control the persons that can access to the data in order to have a secure service.

During the process of sending data to a mHealth cloud, when there is a failure, the stored data can be corrupted (the loss of data or error in the data). The problems related with errors in the stored data are huge when these data are health data. The life of the patients may depend on this data. The failures can happen in different moments: when the data is sent from the device to the cloud, when the data is send from the cloud to the consulter (GP, caregiver, etc) or even when the data is sent from one cloud to another (in the case of systems with several clouds). Now we present some scenarios where failures in data are critical to expose the need of a robust protocol for secure communication in mHealth, If there are errors in the sent data, some false alarms may be activated, which suppose economic losses (e.g. send ambulance to a patient house) and could also suppose that those resources (ambulance, sanitary personal of hospital, etc.) may not be available for other real alarms. In the opposite case, if data must activate an alarm, but due to the failure this alarm is not generated, the life of the patient is endangered. In case of errors when a pharmacist access to the prescriptions given by a doctor, a wrong medicament can be sold to the patients, so their life can be put in risk. In all the cases, any error in the data transmission will put in risk human life, so it is very important to use an optimal protocol to avoid those failures.

Nowadays, peer to peer (P2P) systems have evolved hugely. They use replication schemes [29] and even robust structures [30] to provide fault tolerance and high availability. The use of these structures in public P2P file sharing networks is also common because they are capable of supporting many users and data traffic [31]. In this paper, we propose a secure hierarchical architecture that implements P2P connections between clouds focused on eHealth and mHealth systems. The hierarchic architecture consists of two layers composed by nodes with different roles. The proposed system is based on one of our previous work [32], where we proposed an architecture and protocol for intercloud communication in order to allow sharing resources, services and data between mHealth clouds. In this new work, the secure protocol includes the functions of node discovery, authentication, fault tolerance and an algorithm for network recovery after a network failure. Finally, we provide the performance results in a controlled test bench. Our architecture is highly scalable and allows adding new mHealth clouds and nodes without problems, while maintaining the balanced load in the cloud communication.

The rest of the paper is structured as follows. Section 2 describes some works which use cloud computing for eHealth/mHealth systems. Section 3 presents the design of our hierarchical architecture, the network roles and their operation. Our protocol description and its main functions jointly with the fault tolerance algorithms are explained in Section 4. Section 5 details the security system. Section 6 provides several test bench results obtained in two different scenarios. Finally, Section 7 gives our conclusion and future work.

2. Related work

There are several types of uses of cloud computing for health services. There are studies that show the benefits obtained when using eHealth/mHealth. In this section, we review frameworks that include the use of cloud for improved eHealth/mHealth services, architecture designs for a personal eHealth record that integrate services from cloud computing, the design of eHealth platforms and systems that operate in the cloud.

Chowdhury [33], Ghosh et al. [34], Donahue [35], Deng et al. [36] and Sultan [37] suggested the use of cloud computing to improve eHealth/mHealth services. Chowdhury states in [33] that the cloud scalability, adaptability, cost reduction, and high performance features improve the medical services. He makes a classification of health care systems with cloud computing and makes a comparative study of the discussed papers. On the other hand, Gosh et al. [34] covered multiple topics, besides those mentioned by Chowdhury, they include security aspects and data analytics, as well as the service composition of healthcare services. Moreover, they identified some research topics that have not been properly analyzed till that date by the research community such as the availability, resiliency, and service-level agreements (SLAs) of healthcare cloud. Donahue presents in [35] the benefits when using cloud computing in eHealth to share the user information among health professionals. He says that the information inside a health Cloud can also be better analyzed and tracked (with the proper information governance) so that data on treatments, costs, performance, and effectiveness studies can be analyzed and acted upon. Facilitated by health clouds, patient information can be shared among authorized physicians and hospitals, providing more timely access to life-saving information and reducing the need of duplicate testing. Deng et al. [36] propose a cloud computing application for home healthcare and they address its security and privacy challenges. First, they investigate a number of use case scenarios for home healthcare and they proposed a cloud-based home healthcare architecture. They described a comprehensive methodology for integrating security and privacy engineering process into the

development lifecycle of the proposed cloud-based home healthcare system. Moreover, they sketched out a functional infrastructure plan to integrate the proposed home healthcare application architecture into a commodity cloud. Finally, they discussed a mitigation solution, with a focus on facilitating patient - and data - centric protection for electronic health data, and highlighted its main components that allow patients to encrypt their own data (health or physical activity) prior to uploading the data to the cloud. They say that it is needed further work to develop trust protocols, in order to elaborate the proposed mitigation solution and to address the identified security and privacy requirements thoroughly. Sultan [37] makes a study about the cloud computing development and explore the potential (and future) of contributing to the advancement of healthcare provision. He stated that it offers several advantages to potential users such as "metered" use (i.e., pay-as-you-go) which offers scalability, online delivery of software and virtual hardware services (e.g., collaboration programs, virtual servers, and virtual storage devices) which would enable organizations to obviate the need to own, maintain and update their software and hardware infrastructures.

Now, we will show some frameworks that include the use of cloud computing to improve eHealth/mHealth services. Some authors like Nkosi et al. [38], Wang et al. [39], Alamri [40] Constantinescu et al. [41] and Botts et al. [42] suggested different frameworks for improving health services using cloud computing. Nkosi et al. [38] suggest a framework to relieve mobile devices from executing heavier multimedia and security algorithms when delivering mHealth services. The proposed framework uses a cloud computing protocol management model which intends to provide multimedia sensor signal processing and security as a service to mobile devices. They suggest that multimedia and security operations can be performed in the cloud, allowing mobile health service providers to subscribe and extend the capabilities of their mobile health applications beyond the existing mobile device limitations. Wang et al. [39] propose a mobile cloud telemedicine framework and discuss its potential performance by taking advantage of the on-site monitoring capability of Android mobile device and the abundant computing power of the cloud. Alamri [40] proposes a cloud computing framework that uses scalable video coding. It has efficient video compression that relieves heterogeneous devices when they execute heavier multimedia applications and uses database storage and when delivering mobile health services over limited bandwidth resources.

An adaptive video bit rate in multimedia health services comprise desirable key features that should be considered. Constantinescu et al. [41] proposed the SparkMed data integration framework for mobile healthcare (m-Health), which significantly benefits from the enhanced network capabilities of LTE wireless technologies, by enabling a wide range of heterogeneous medical software and database systems (such as the picture archiving and communication systems, hospital information system, and reporting systems) to be dynamically integrated into a cloud-like peer-to-peer multimedia data store. Botts et al. show an example in [42] of an architectural design for a personal e-health record that integrates services from cloud computing. It is an architectural design for a personal health record system called "HealthATM" that utilizes and integrates services from Google's cloud computing environment. These services are integrated into an unobtrusive and easy to use ATM-style interface for health consumers and care providers to manage and track their health.

Finally, we show some platforms and systems to support e-Health service systems capable to operate in Cloud. There are some works like the ones presented by Fan et al. [43], Ruiz-Zafra et al. [44], Nijon et al. [45] and Xia et al. [46] where they propose mobile platforms and systems designed to support eHealth/mHealth systems and are able to operate in the cloud. Fan et al. presented in [43] DACAR (Data Capture and Auto Identification Reference) project, a novel eHealth services platform. Firstly, the DACAR platform facilitates the development of eHealth applications by addressing the most typical requirements, including authentication, authorization, secure data transmission, persistence, integrity, confidentiality and audit trail. Secondly, it provides a suite of hardware and software solutions to integrate the capture, storage and consumption of sensitive medical data. Thirdly, it supports large-scale deployment and delivery of e-Health services using a scalable and cost-effective Cloud infrastructure. Ruiz-Zafra et al. [44] present Zappa, a platform for the development of eHealth/mHealth systems which is able to operate in the cloud. The platform is based on open and close source hardware and open-source software that reduces its acquisition and operation costs. The platform is intended to provide uninterrupted monitoring with the goal of obtaining some information that can be subsequently analyzed by physicians for diagnosing. Nijon et al. [45] present Auditeur, a general-purpose, energy-efficient, and context-aware acoustic event detection platform for smartphones. Auditeur is backed by a cloud service to store user contributed sound clips and to generate an energy-efficient and context-aware classification plan for the phone. Auditeur's energy-aware algorithm is capable of increasing the device-lifetime by 33.4%, sacrificing less than 2% of the maximum achievable accuracy. Xia et al. [46] developed a cloud-based system for real-time ECG monitoring and analysis. They presented a use case, in which ECG data was uploaded to the web server from a mobile phone at a certain frequency and signal analysis was performed in real time after the signal was uploaded to the server. The system has been proven to be functional, accurate and efficient in monitoring and analyzing the ECG data.

Papers included in this section present cloud-based architectures or services, but without any performance study or test. Moreover, they are explained without any technical detail without depth. We also include a fast fault tolerance algorithm, so the system is able to detect topology changes and update the routing tables to ensure the correct routing information.

3. Architecture description

The proposed architecture is based on a logic interconnection of clouds that share the same type of resource. The architecture is based on a hierarchy of two layers: the organization layer (divided into two sub-layers) and the distribution layer [31][47]. Org_Nodes (Organization Nodes) belong to the organization layer, and Dist_Nodes (Distribution Nodes) belong to the distribution layer. Each cloud has an identifier called Cloud_ID (unique in the whole architecture for each cloud, all nodes belonging to this a cloud have the same Cloud_ID) and each node has an identifier called Node_ID (unique in each cloud). This kind of architecture offers greater scalability and robustness against cloud failures [48]. Our architecture uses nodes that form networks in charge of specific services in a medical environment which allow interconnecting the different mHealth clouds. When a node needs to obtain some information, firstly it will try to get it from its network. If no results are found, the node of a cloud tries to search this information from other clouds using our protocol. Once the node has the information, it will act as a cache for its network to share this information. In this section, we present the proposed architecture and the role of nodes that form the different clouds as well as the tasks they have assigned.

3.1 Roles of each node.

To work properly, we have defined 3 types roles for the nodes, keeping in mind that a node could run all them simultaneously.

- Dist_Node: These nodes have connections with Dist_Nodes from other clouds. Dist_Nodes are used to send searches and data between clouds following a hub-and-spoke distribution paradigm where all traffic moves along Dist_Nodes connected to the hub at the center as a star network. A Figure 1(a) shows the state diagram of a Dist_Node Node. IT leaves the inactive state when it tries to establish a connection with other nodes. Once connectivity is established, it performs the data exchange and remains in active status while it has information to transmit.
- Org_Node_Level-1: These nodes organize Dist_Nodes of the same cloud in zones to have a scalable architecture. As we can see in Figure 1(b), when it has started, the Org_Node_Level-1 should calculate the distance and cost to arrive to the rest of nodes in the Org_Node_Level-1 network. Being in active status, the Org_Node_Level-1 can receive data or recalculate the routes.
- Org_Node_Level-2: These nodes have connections with Org_Node_Level-2 from other Clouds. These nodes organize connections between Dist_Nodes from different networks. Figure 1(c) shows that the diagram state of Org_Node_Level-2 nodes. Before reaching the active status, they have to to calculate the distance and cost to arrive to the rest of Org_Node_Level-2 nodes. Org_Node_Level-2 nodes are also Org_Node_Level-1 nodes.

3.2 Proposed architecture and its operation

Our architecture is composed by two main levels or layers, distribution layer and organization layer (although organization layer is divided into two sub-levels). The organization layer is the one who has fewer nodes. Org_Nodes maintain and manage the proposed interconnection architecture. They help to establish connections between different Dist_Nodes from different clouds. All Org_Nodes in a cloud have the same Cloud_ID. For security reasons, an Org_Node only knows the Node_ID of the other Org_Nodes. An Org_Node does not know the network layer address of all Org_Nodes located in the architecture for perpetrating malicious attacks. The SPF routing algorithm can route information between Org_Nodes quickly and allows fast searches for Dist_Nodes to obtain, as soon as possible, its connections to other Dist_Nodes from other clouds.

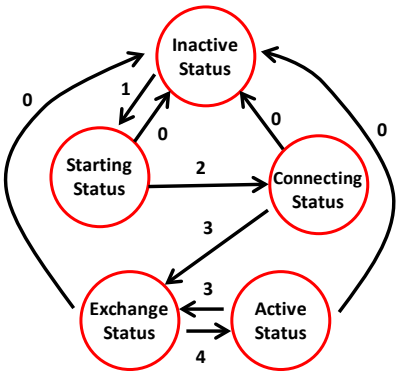

Figure 1 (a). Diagram State of a Dist_Node

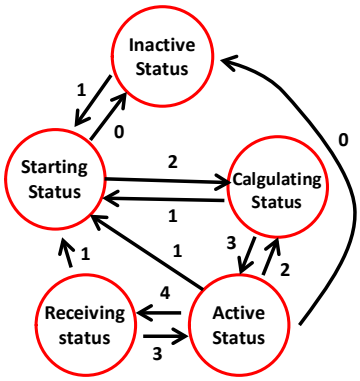

Figure 1 (b). Diagram State of aOrg_Node_Level-1

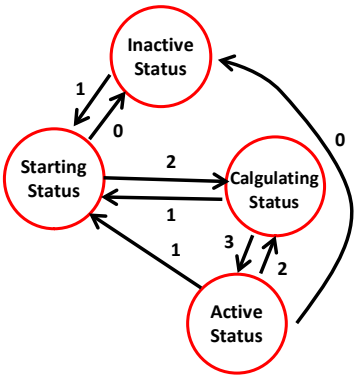

Figure 1 (c). Diagram State of a Org_Node_Level-2

The organization layer has two layers of routing in order to have a more scalability. The level 1 layer, composed by Org_Nodes_Level-1, allows routing information within the cloud. The identifier for routing information is the Node_ID. The Level 2 layer is formed by Org_Nodes_Level-2 and allows routing between different clouds. The identifier that allows routing information is the Cloud_ID. In both layers, each Org_Node locally runs the routing algorithm for all Org_Nodes network to build its shortest paths tree (SPT) for each Org_Node. Then, the Org_Node chooses the path with the smallest metric. Org_Nodes_Level-1 does not directly reach other clouds. Therefore, any information which destination is different to an Org_Nodes_Level-1 of its own cloud must be sent to the closest Org_Nodes_Level-2.

Dist_Nodes are used to transmit data between pairs of different clouds. They establish adjacencies directly with Dist_Nodes from other clouds following the spoke-hub distribution paradigm. Each new Dist_Node must send a request to its Org_Node_Level-1 to request such adjacencies. The Org_Node_Level-1 forwards this query to the Org_Node_Level-2 in order to reach other clouds. Org_Node_Level-2 will select Org_Node_Level-2 nodes from the destination clouds and will send them the query. Org_Node_Level-2 nodes from other clouds will send the query to their Org_Node_Level-1 in the Cloud. Org_Node_Level-1 selects the best available Dist_nodes based on their adjacencies and their available loads. Then, it will send this entry to the Org_Nodes_Level-2 in its cloud. Org_Nodes_Level-2 will transmit them to the initial Org_Nodes_Level-2. In order to establish the connections, Org_Nodes_Level-1 of the source cloud will forward the information about the selected Dist_Nodes adjacencies to the applicant Dist_Node. The applicant Dist_Node will get two selected Dist_Nodes from all clouds. Finally, using the information obtained, the Dist_Node will attempt to establish adjacencies with the received Dist_Nodes. Then, they will be added to the distribution table of Dist_Nodes.

Figure 2 shows an example of the network when the proposed architecture has converged.

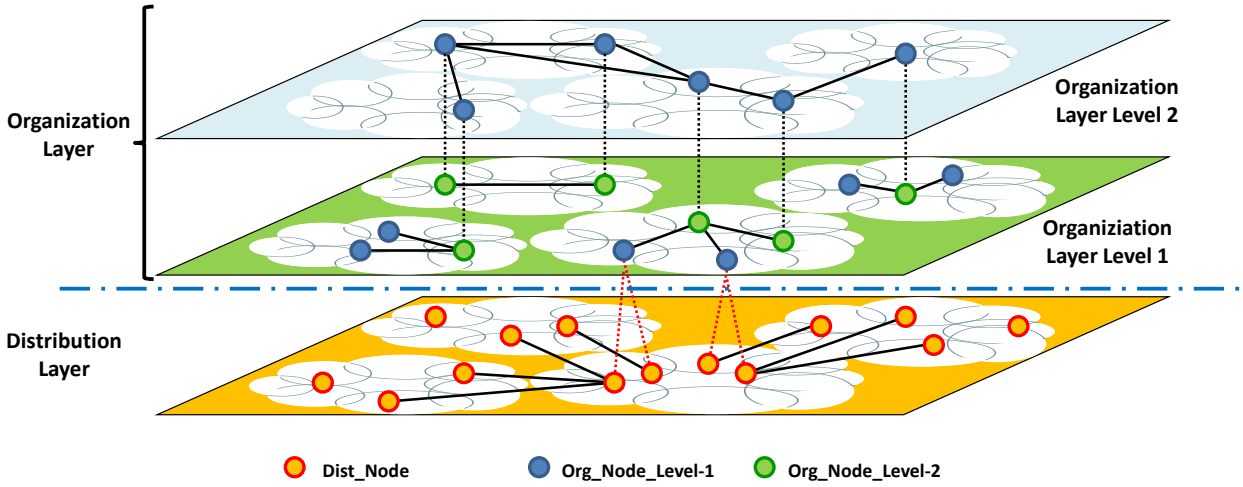

Figure 2. Proposed architecture

To get a good response to network failures, two important parameters are defined:

- The first parameter is the promotion parameter (δ). It allows establishing an order in the Org_Nodes layer. This parameter depends on the bandwidth of the node and its age in the system. It is used to know which node should have a greater role. Nodes with higher bandwidth and more time in the cloud will have higher δ, so they are preferred to promote.
- The second parameter is the capacity (λ). It allows to any Org_Node of our architecture to know which nodes have the most available capacity and establish connections with them. It depends on the node bandwidth (in kbps), the number of available connections, the maximum number of connections and their % of available load.

The main task of Org_Nodes is to maintain the stability of the different clouds and manage the interconnection between clouds within the architecture. When a new Org_Node wants to be part of a cloud, it must authenticate with other Org_Nodes of its cloud. The new node should share its information with the rest of the cloud and it starts with its upstream and downstream bandwidth, its maximum number of supported connections from other nodes and its maximum % of CPU load used for joining the architecture. The first node in the cloud will be labeled as Node_ID = 1 and the system will assign the Node_IDs sequentially as new nodes are added to the cloud. The first time a node joins the architecture, it must choose a Cloud_ID which consequently implies the choice of cloud to which it binds.

The amount of Dist_Nodes and Org_Nodes depend on the number of devices in the cloud, the number of clouds and their data traffic. When there is a node failure δ parameter is used to know which node should promote. A regular promotion path is from Dist_Node to Org_Node_Level-1 and from Org_Node_Level-1 to Org_Node_Level-2. The node must authentic ate in each step.

As routing protocol, we have chosen the Shortest Path First (SPF) routing algorithm to route information between Org_Nodes and between Dist_Nodes [49]. This routing algorithm is faster to find adjacent Dist_Nodes. However, other routing protocols can be used. Org_Node_Level-l uses the Node_ID parameter to route the information inside its cloud while Org_Node_Level-2 is in charge of routing the information between clouds. For this reason, Org_Node_Level-2 uses the Cloud_ID to send data in its layer. Org_Node_Level-l nodes only add Org_Node_Level-1 nodes in their clouds. An Org_Node_Level-2 will add all Org_Node_Level-1 nodes in its cloud because it is also Org_Node_Level-1. Finally, each node runs the SPF algorithm to know the best path to reach a destination. The metric for each route is based on the number of hops to a destination and the virtual-link cost of those nodes involved in the path, which is based on node's capacity (λ). The more is the node's capacity the lower is its cost.

4. Protocol description and fault tolerance procedures

In the architecture operation, there are many different situations such as the incorporation of a new cloud and the addition of new nodes into existing clouds. This section explains the operation of the protocol in these situations.

When a change occurs in the network topology, it is important to distinguish between two situations. On the one hand, topology change may have been produced by the voluntary abandonment of the node or because a node failure. In the first case, the node who wants to leave the network can announce it to the rest of the network, while in the second case, the network detects itself the situation. In our design we let the network discover in both cases that a node is not yet available. Our design does not include backup nodes to avoid overloading the network with control messages and keeping nodes replicating the network information, but it can be included as we did in [47].

4.1 The entry of a new cloud

As Figure 3 shows, when the first no of a new cloud joins the architecture (it can be previously configured as a node of a new cloud or it can discover that it is the first node of that cloud because it gets no reply from other node from the same cloud), it sends a discovery message with its Cloud_ID to find Org_Nodes_Level-2 from other clouds, which are known in advance. Org_Node_Level-2 nodes from other clouds will reply with their Cloud_ID and their λ parameter. If no reply is received after a period of time, the node will begin the process again. The node of the new cloud will choose the Org_Node_Level-2 nodes with highest λ and will send them a connection message. This connection will be acknowledged (ACK) with a welcome message containing their neighbor list. Existing Org_Node_Level-2 nodes will add this new entry to their topological database and will recalculate routes using SPF algorithm. Finally, Org_Node_Level-2 nodes will send periodic keepalive messages to indicate that they are still alive. If a neighbor does not send the keepalive message for a dead time, this entry will be erased from its database.

4.2 New Dist_Node joins

When a new node tries to join the architecture, it will be Dist_Node. Firstly, it will send a discovery message with its Cloud_ID to the Org_Nodes_Level-1, which are known in advance or by bootstrapping [50]. Only the Org_Node_Level-1 with the same Cloud_ID will reply with their λ parameter. The Dist_Node should wait for a hold time and choose the Org_Node_Level-1 with highest λ. The Dist_Node will send the discovery message again if there is no reply for a hold time. When a Dist_Node receives the acknowledge discovery message it will send a connection message to the selected Org_Node_Level-1. Org_Node_Level-1nodes will reply with a welcome message with the assigned Node_ID and the information related with the backup Org_Node_Level-1 and this information is added to its Dist_Node's table. Finally, Dist_Node will send keepalive messages periodically to the Org_Nodes_Level-1 in order to inform about its status. The procedure and message exchange is shown in Figure 3.

4.3 New Org_Node_Level-1

When an Org_Node_Level-1 fails down or leaves the network, the first node that will notice it will be its closest Org_Node_Level-2. Then, it will have to look for the most appropriate new Org_Node_Level-1. In order to do it, the Org_Node_Level-2 will send a request to all Org_Node_Level-1 nodes. They will reply with the Node_ID of the Dist_Nodes with the highest δ in its Dist_Node table. Org_Node_Level-2 will process the received replies in order to select the Dist_Node with highest δ. After that, Org_Node_Level-2 sends a message to the Org_Nodes_Level-1 with the highest δ, which will forward the message to the selected Dist_Node. Dist_Node will become Org_Node_Level-1 and will send a disconnection message to its Org_Node_Level-1. If the Org_Nodes_Level-2 does not receive changes for a hold time, it will send a new request message to the second Dist_Node with highest δ. This process will continue till if finds a new Org_Node_Level-1. Message exchange is also shown in Figure 3.

4.4 New Org_Node_Level-2

When an Org_Node_Layer-2 fails down or leaves the network, Org_Node_Layer-1 neighbors in its cloud will detect it because the lack of keepalive messages, so they will update their Org_Node_Layer-1 database and will propagate this information through the Org_Node_Layer-1 network. The Org_Node_Layer-1 with highest δ will receive this update and it will become Org_Node_Level-2. Then, it will send a message to the Org_Node_Level-2 nodes in its cloud in order to inform them about its new role, and to request them information about the Org_Node_Layer-2 network. In case of not being any other Org_Node_Layer-2 in its network, it will send a discovery message to the Org_Nodes_Level-2 nodes, which are known in advance. The new Org_Nodes_Level-2 will follow the procedure explained in Figure 3.

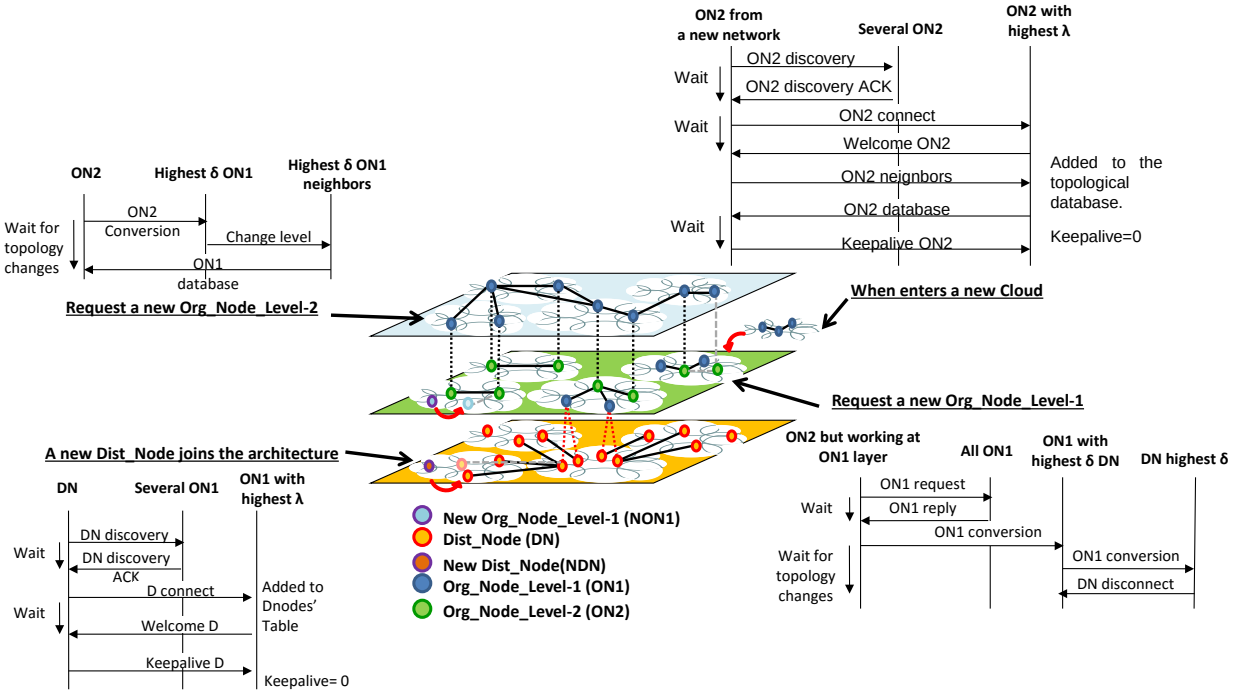

Figure 3. Main tasks in our architecture

5. Security procedure

This section shows the designed security procedure to secure the data transmission across the network and the authenticity of the identity of the nodes belonging to different clouds.

To ensure that new nodes requesting to join the network are reliable, the system must verify their identity and that this identity is not in use on the network. To perform this process, the system will use an authentication protocol. The process is slightly different if the requester node is a Dist_Node or an Org_Node. A new Dist_Node will send a discovery message to the Org_Node_Level-1 nodes in order to request to join the architecture. Once this connection is stablished Dist_Nodes will send an authentication message with its Node_ID. Org_Node_Level-1 nodes will forward the message to the Org_Nodes_level-2, which will check if there is any other node using this identity. If the checking is correct, the connection is confirmed but not yet accepted. Then, Org_Node_level-1 asks the closest Org_Node_level-2 about the authentication procedure. Then, it will request the password to join the network. Org_Node_level-2 will request the password to the new Dist_Node Finally, Org_Nodes_level-2 accept or deny the connection as a function of the received reply. If the password is right, the new Dist_Node will be able to exchange information with the rest of network. However, if password was wrong, this node will be tagged as dangerous and it will not be connected to the network. Figure 4 shows this process. On the other hand, when the connection request is generated by an Org_Nodes_level-1 or Org_Nodes_level-2, the process is simpler (See Figure 5). In this case, the message exchange is performed directly between the requesting node and Org_Nodes_level-2. In both cases, after accepting the connection request, nodes will update their neighbor and routing table.

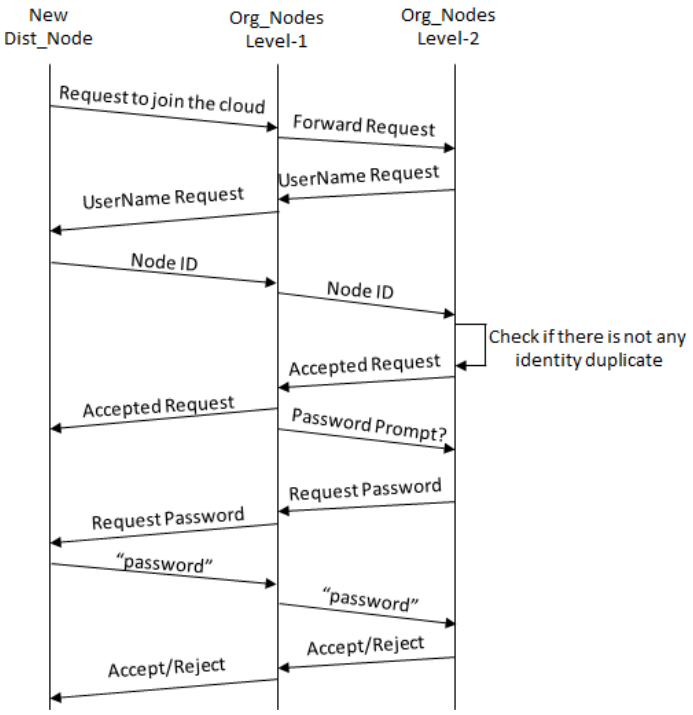

Figure 4. Authenticating process for a new Dist_Node

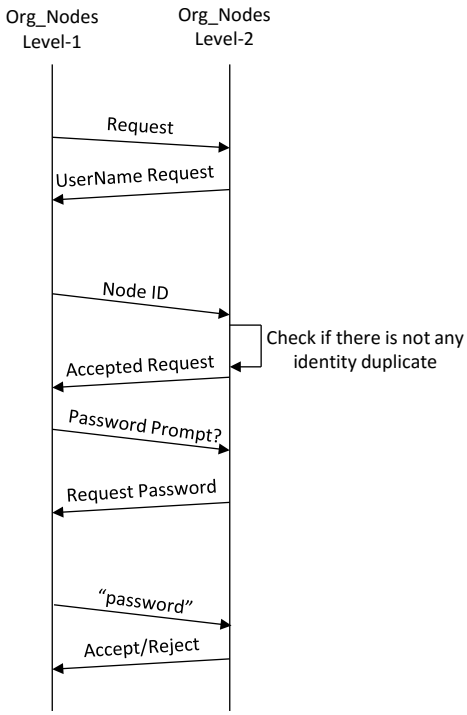

Figure 5. Authenticating process for a new Org_Node_Level-1

Finally, it is important to ensure that the sender of a message is reliable and that the content of the message does not contain dangerous information and altered data that could compromise the information of the network. For this reason, we propose the use of a cryptographic hash function. We have selected the Secure Hash Algorithm (SHA-1) [51] [52]. SHA-1 is a mathematical algorithm that transforms a block of data in a line of characters of a fixed length. No matter how much data is inserted, the result will be always a hash of the same length. SHA-1 is frequently used due to its stability between its speed and its security level. Its performance is maintained in low computing devices because its operation mainly depends on the processor. Figure 6 shows the process undertaken to digitally encrypt our data and how decrypt them the on destination.

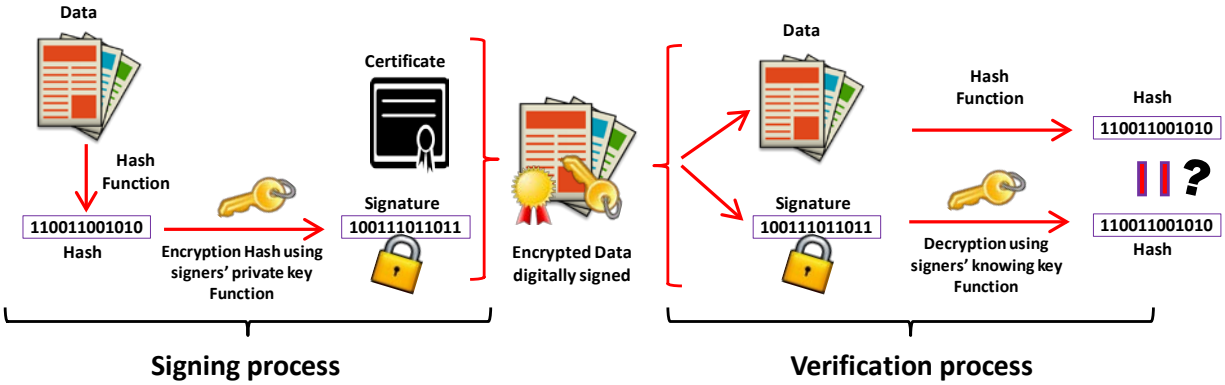

Figure 6. Process of singing, encrypting and decrypting the data using hash and private key function

The data are processed using a hash and they are digitally signed to prevent its alteration. After this, the information is ready to be transmitted. SHA-1 converts a text string in 40 characters with no matter of the length of the original string. SHA-1 has no reverse method to obtain the original information. In order to verify that the received information is correct, the destination node compares the received has with the hash obtained using its has function. The value of the key function is pre-shared between the nodes members of the network.

Figure 7 shows the block diagram of our secure algorithm for data transmission. When a Dist_Node receives the request to send it a message, the Dist _Node checks the identity of this sender to ensure the reliability of this node. If the verification is successful the system proceeds to receive the message and checks if the hash is correct. If the received and the created hash are mismatched, the Dist_Node sends a message to its Org_Node_Level-1 to inform about it. If it happens more than one time, Org_Node_Level-1 will send a broadcast message to the rest of network.

When the sender identity verification is wrong, the system sends a request to receive the updated information about the sender identity to its Org_Node_Level-1. The Org_Node_Level-1 analyzes the sender and cloud identity. If this verification is successful, Org_Node_Level-1 sends the updated data to the Dist_Node requesting this information. If the Org_Node_Level-1 does not know the sender, it will send a request identity message to the rest of Org_Node_Level-1 nodes (including those that are also Org_Node_Level-2 nodes) asking for the identity of this sender. If nobody knows the identity of sender, the Org_Node_Level-1 who had connection with the Dist_Node will generate a broadcast message to the whole network informing about the threat.

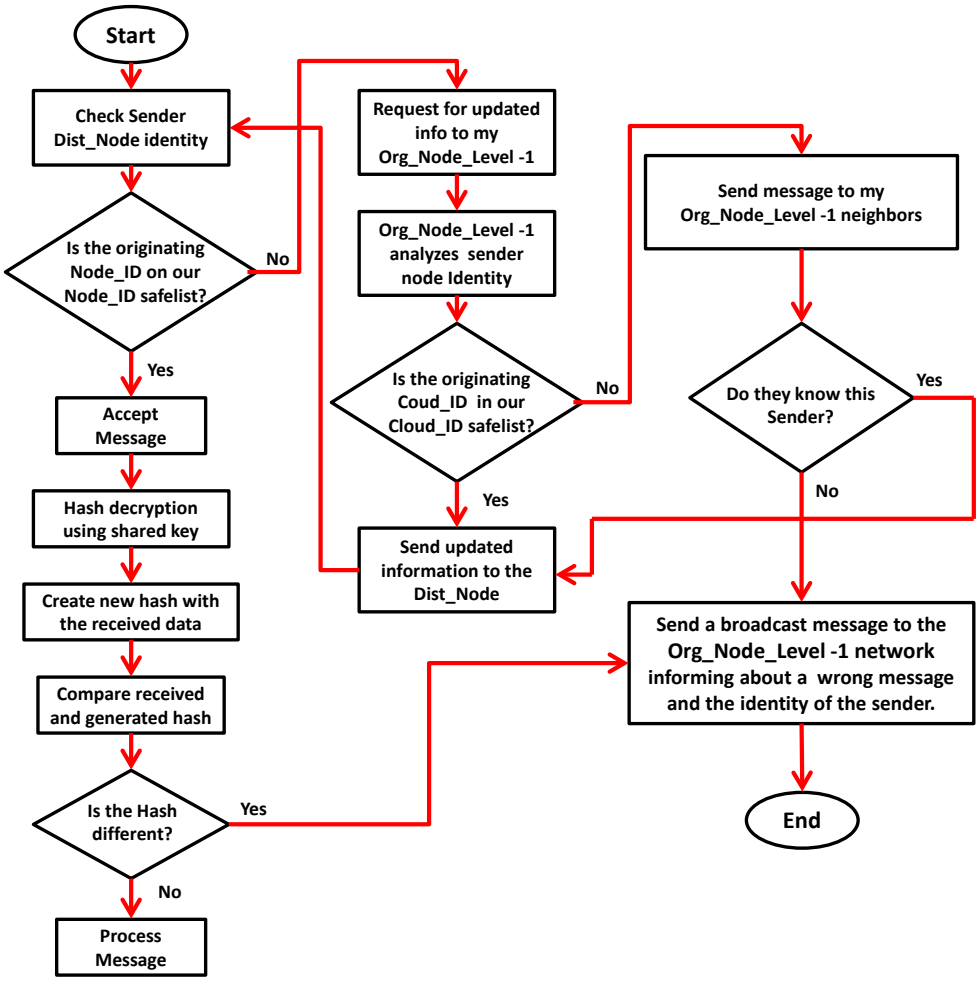

Figure 7. Secure system for data transmission.

6 Test bench results

In order to test the efficiency of our system and the stability of the proposed architecture, we performed a set of real experiments using two different scenarios. They allow us to know the time needed by the proposed architecture to converge when a node fails down or when a node is removed from the architecture because the security is compromised. This section shows the scenarios used in each test and the results obtained in each case, in terms of number of messages, number of Bytes and broadcast messages detected through the network.

The first scenario is shown in Figure 8. It has 2 clouds where each cloud has a single Org_Node_Level-2, which is also an Org_Node_Level-1. There are 11 Dist_Nodes in each cloud. In order to test the impact of removing a single node, we start with a topology where a Dist_Node from one cloud has connections with 8 Dist_Nodes of the other cloud.

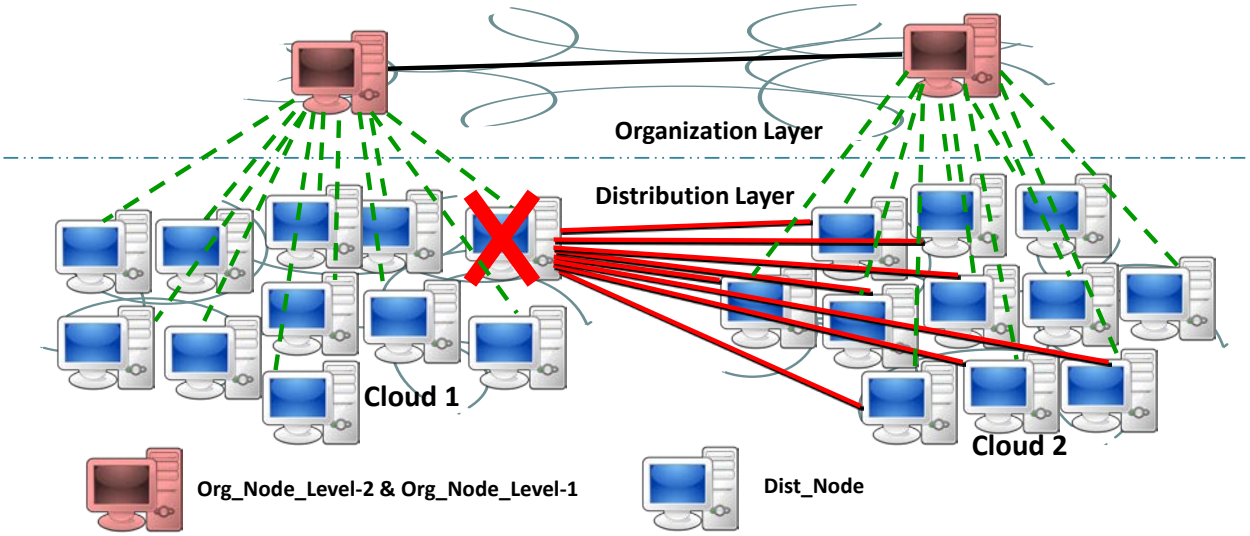

Figure 8. Scenario 1.

The scenario 2 has the topology shown in Figure 9. There are 3 clouds. Each cloud has a single Org_Node_Level-2, which is also an Org_Node_Level-1. One Dist_Node of clud 1 has connections to 7 Dist_Nodes of cloud 2 and 6 connections with Dist_Nodes of cloud 3, so the Dist_Node of Cloud 1 has 13 connections with other Dist_Nodes.

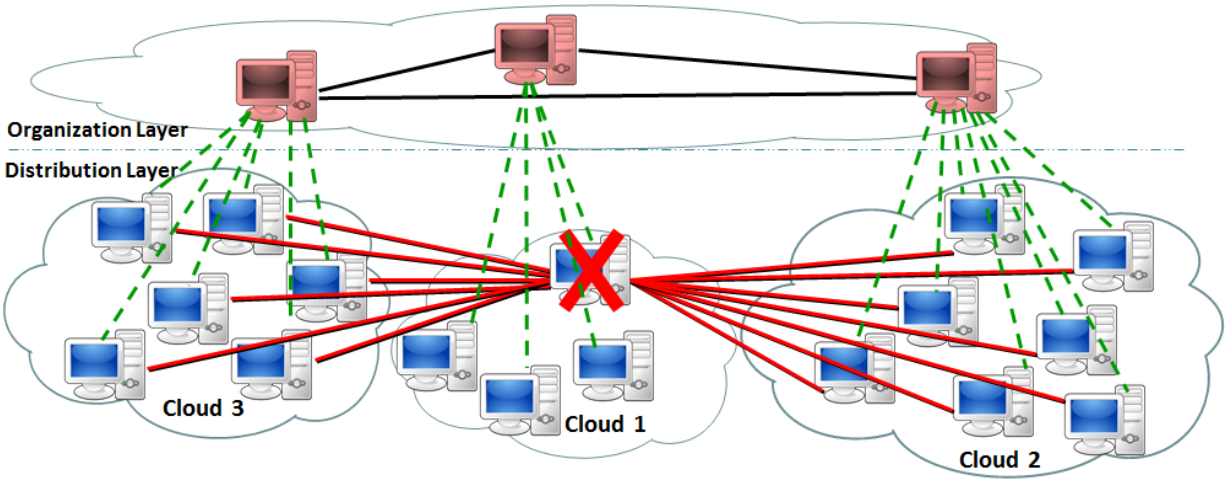

Figure 9. Scenario 2.

The parameters used for the first and second test of each scenario are shown in Table 1. The default parameters are used for the first test and the modified parameters for the second test.

Parameter	Default parameters	Modified parameters
Downstream bandwidth:	1024 Kbps	1024 Kbps
Upstream bandwidth:	256 kbps	256 kbps
Time of keepalive:	20 Seconds	30 seconds
Value of holdtime:	60 Seconds	60 seconds
Max. % of used CPU:	50 %	50 %
Timer:	4 Seconds	10 seconds

Table 1. Parameter values for the tests performed in scenario 1 and scenario 2.

6.1 Test bench of scenario 1 using the default parameters

Once the whole network has converged and keepalive messages between nodes are exchanged, we remove from the cloud 1 an unsecure Dist_Node, which has a connection with 8 Dist_Node from cloud 2. Then, the designed algorithm and protocol is started for selecting a new Dist_Node of this cloud to have connections with Dist_Nodes from other clouds. The architecture uses the default parameters. Our purpose is to measure the impact to the network when the security is applied.

Figure 10 shows the number of Bytes in the network when it converges. As we can see, there are peaks higher than 6000 Bytes every 15 seconds. We observed 3 peaks higher than 10000 Bytes at second 50, second 105 and second 340. The rest of the time, the number Bytes obtained ranges is between 2000 and 4000. The average value of Bytes is around 4500 Bytes.

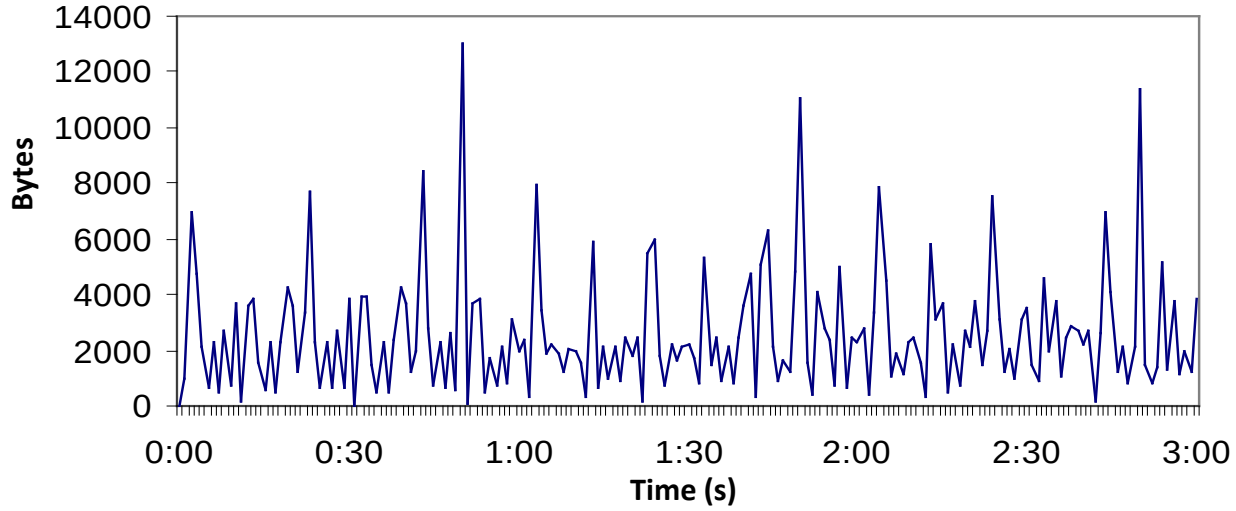

Figure 10. Number of Bytes obtained when the network converges in Scenario 1, using default parameters.

Figure 11 shows the number of registered messages when the network converges. Every 25 seconds the number of messages increase till a value over 70-80 messages. During the rest of the time, the network had a variable number of messages which ranges around the 40-50 messages.

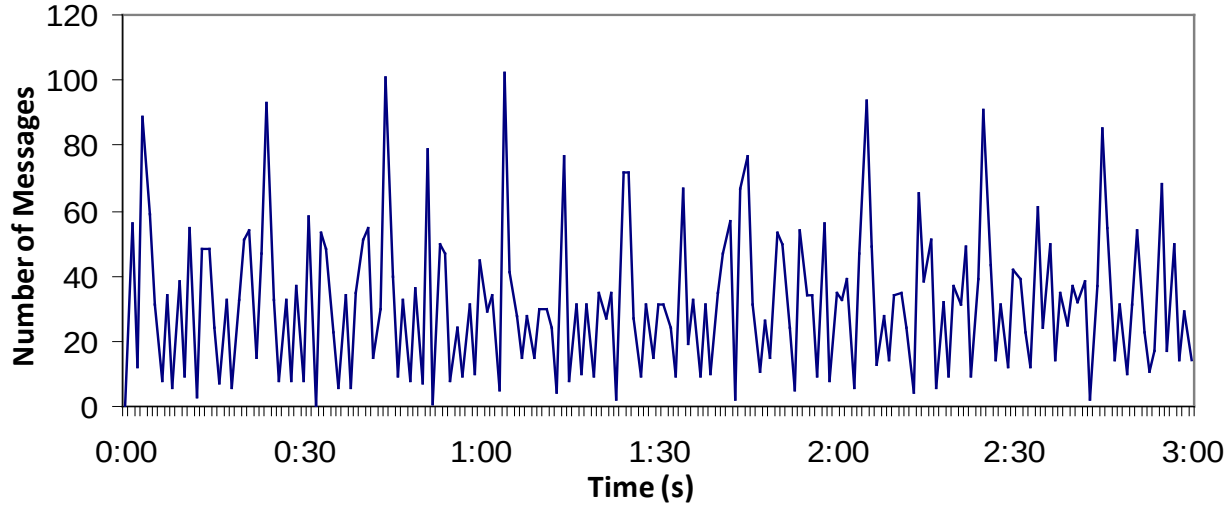

Figure 11. Number of messages obtained when the network converges in Scenario 1, using by default parameters.

Finally, Figure 12 shows the number of broadcast messages sent through the network in the first scenario when the default parameters. In this case, we can see that the fist messages are registered after a minute. From the second

105, the network registers 2 messages every 5-6 seconds. These messages are mainly used to update the routing tables of different nodes. For this reason, when the entire network knows that a failure has happened, it starts to inform the rest of the nodes of the network and update their routing tables.

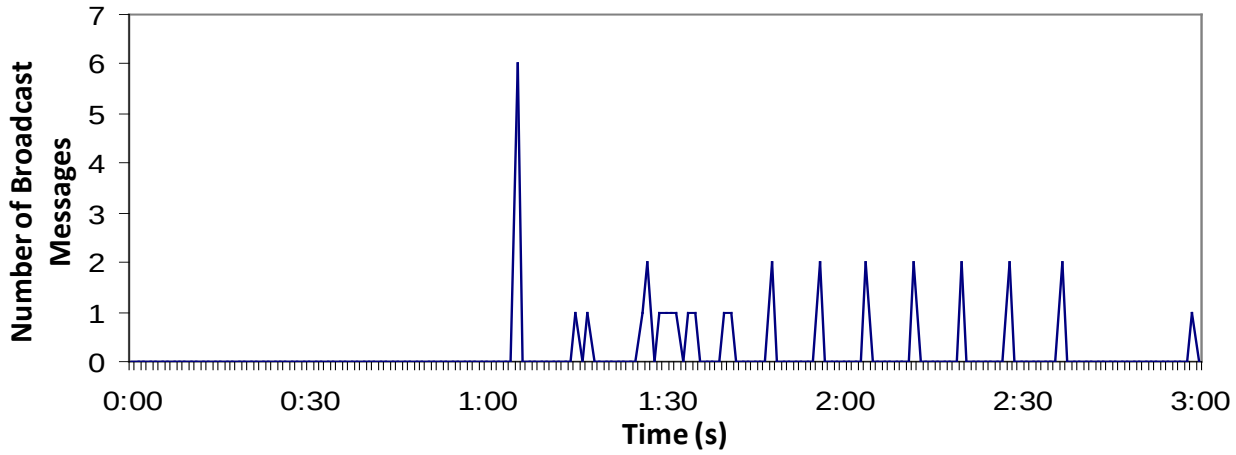

Figure 12. Number of broadcast messages when network converges in Scenario 1, using default parameters.

6.2 Test bench of scenario 1 using modified parameters

In this second part of the test, we used the scenario 1 and the same procedure to remove the unsecure node, but in this case we used the modified parameters of table 1.

Figure 13 shows the number of Bytes obtained when the network converges after removing an unsecure node. If we compare these results with those obtained in Figure 10, we can see that the value of the peaks is lower, although they are also periodically generated approximately every 15 seconds. The average value of the number of Bytes is around 3100 Bytes.

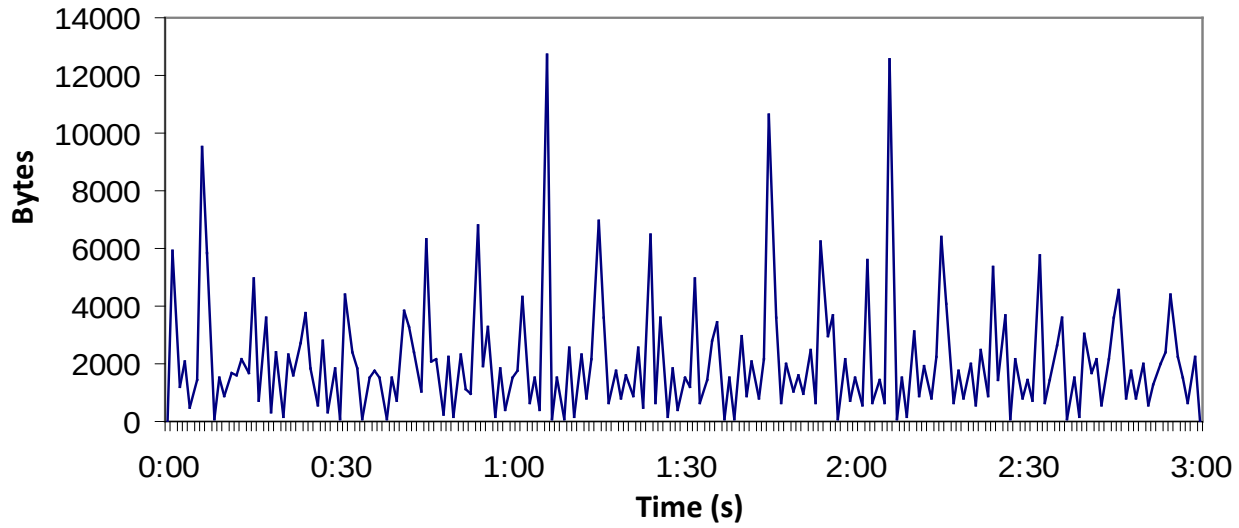

Figure 13. Number of Bytes when the network converges in Scenario 1, using modified parameters.

Figure 14 shows the number of messages obtained when the network converges, using the modified parameters. We observe that the number of peaks generated has values over 50 messages. The rest of the time, the network has a variable number of messages of about 30 messages. Comparing the results to those obtained in Figure 11, the number of messages than when we use default parameters, which had (around 40-50 messages).

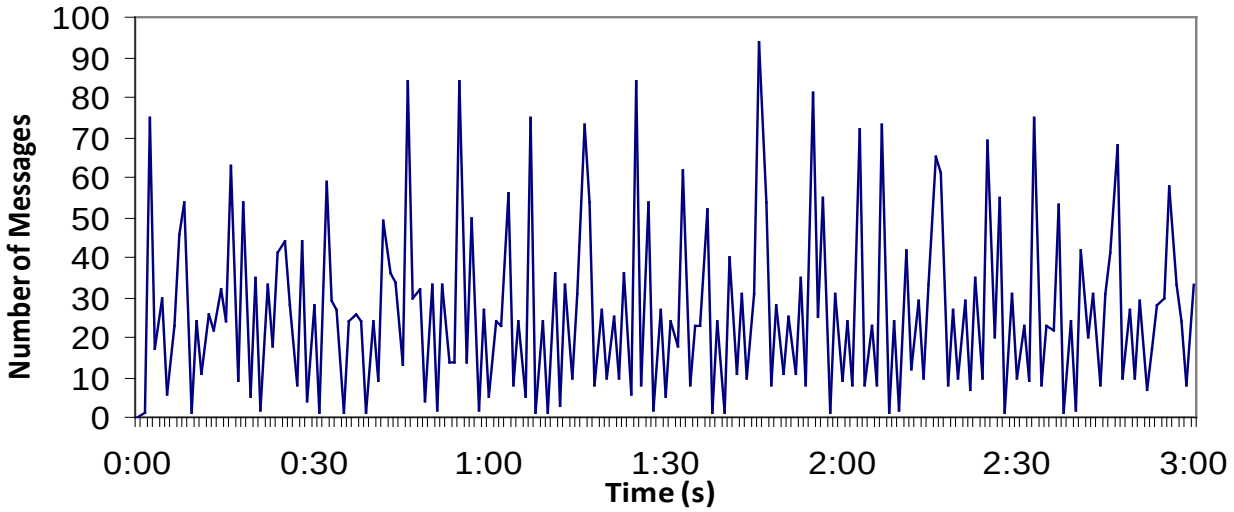

Figure 14. Number of messages observed when the network converges in Scenario 1, using modified parameters.

Finally, Figure 15 shows the number of broadcast messages sent through the network in Scenario 1 using the modified parameters. In this case, the first broadcast messages are obtained just few seconds after initiating the network convergence. When it is compared with figure 12, in this case, the network registers higher number of broadcast messages and this is because the network is aware of removing this node much faster than when it uses the default parameters. For this reason, the network starts sending broadcast messages earlier. The rest of the time, the number of broadcast messages is around 2, but does not follow a periodic pattern.

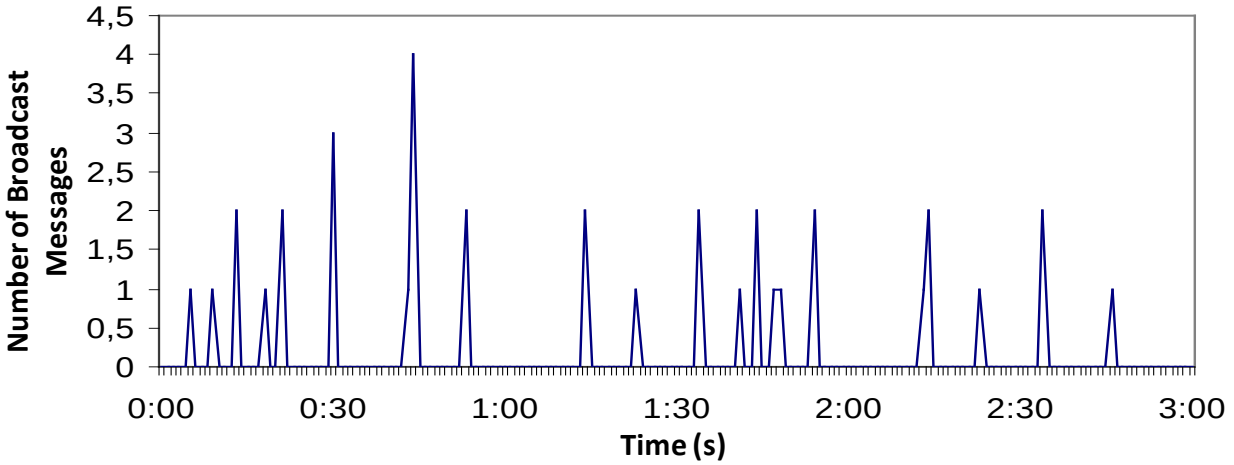

Figure 15. Number of broadcast messages observed when network converges in Scenario 1, using modified parameters.

6.3 Test bench of scenario 2 using the default parameters

In this case we performed the test using scenario 2.Once the whole topology has converged and keepalive messages are received, we remove the Dist_Node because it became suddenly unsecure. Our purpose is to test the recovery algorithm when a security threat happens. We used the default parameters shown in Table 1.

Figure 16 shows the number of Bytes when the network converges, using default parameters. We observed only 3 peaks of about 12000 Bytes. The rest of the time we observed several peaks with values around 3000 and 6000 with an average value of about 4500 Bytes.

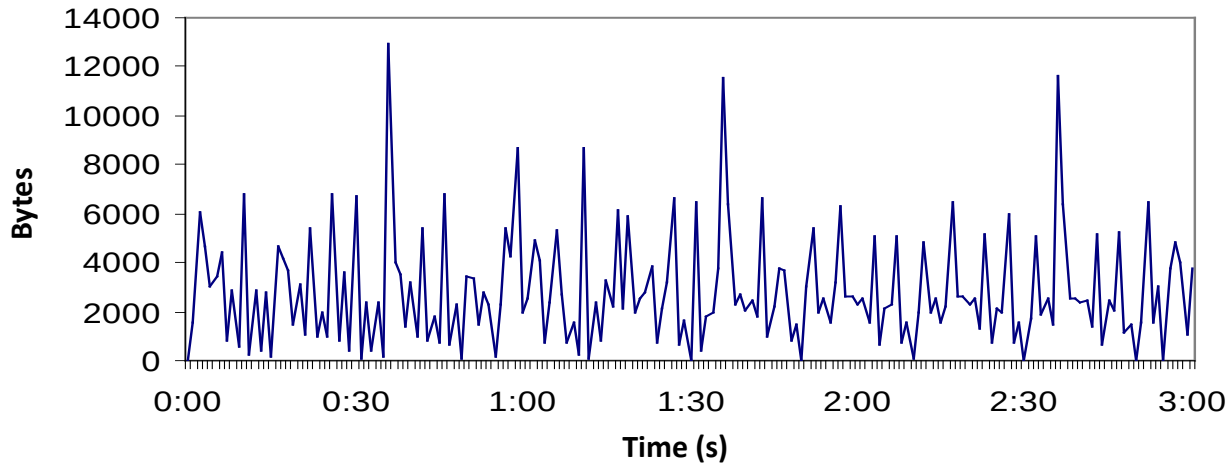

Figure 16. Number of Bytes observed when the network converges in Scenario 2, using default parameters.

Figure 17 shows the number of messages when the network converges using default parameters. We observed that the number of messages varies very much (between 20 and 100). These variations generate peaks every 5-6 seconds. The average number of messages is around 50 messages.

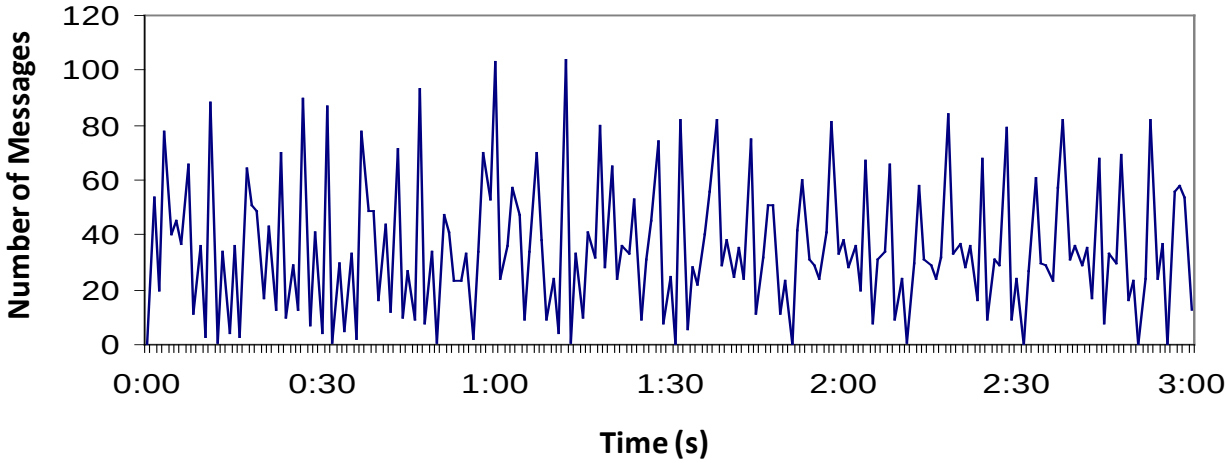

Figure 17. Number of messages observed when the network converges in Scenario 2, using default parameters.

Finally, Figure 18 shows the number of broadcast messages observed in the network in scenario 2 using the default parameters. In this case, the first broadcast messages are obtained just few seconds after the network convergence has started. In most cases, the number of broadcast messages is around 2 in exception of a peak registered at second 160, with a value of 9 broadcast messages.

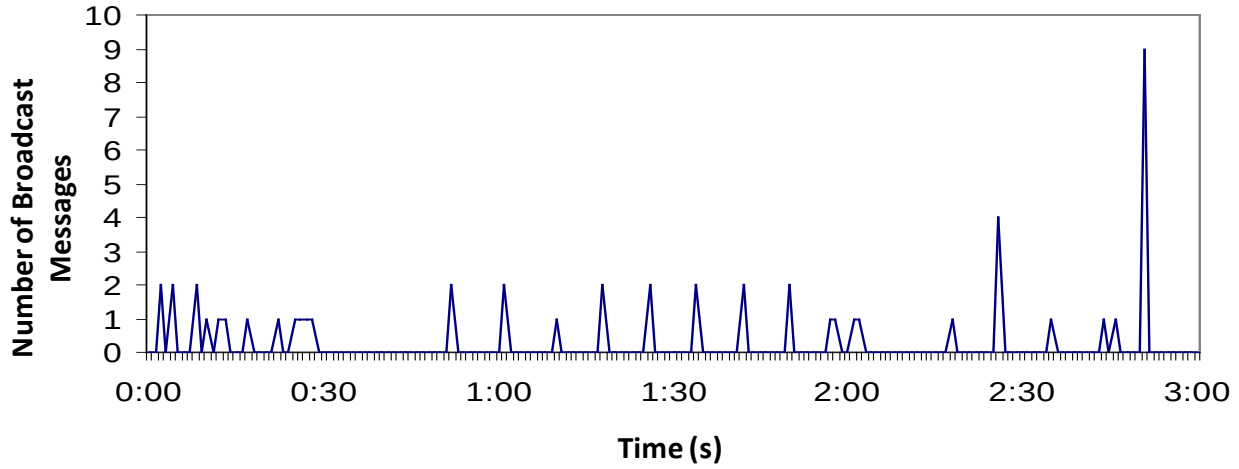

Figure 18. Number of broadcast messages observed when network converges in Scenario 2, using default parameters.

6.4. Test bench of scenario 2 using modified parameters

In this case we take measurements of scenario 2 using the modified parameters shown in table 1.

Figure 19 shows the number of Bytes obtained when the network converges after a node has been removed from the network for security reasons. If we compare these results with those obtained in Figure 16, we can see that there are 2 peaks with a value around 14000 Bytes and a peak at the beginning of the test with a value of 30000 Bytes, the rest of time, the amount of Bytes is between 4000 and 5000 Bytes. In fact, the average value of messages (around 4000 messages) is slightly smaller than the value registered when default parameters are used in scenario 2 (Figure 16).

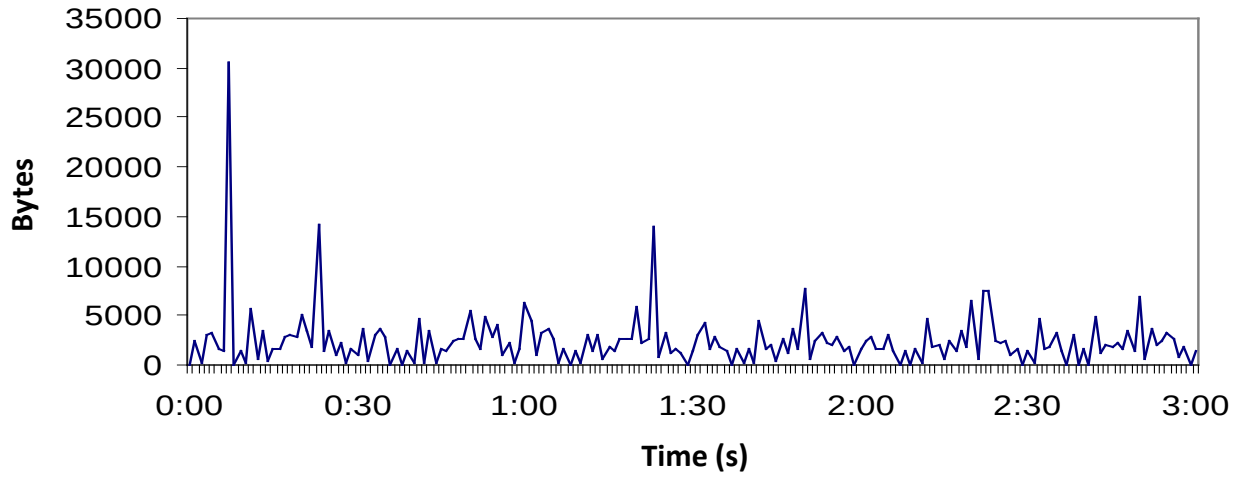

Figure 19. Number of Bytes observed when the network converges in scenario 2, using modified parameters.

Figure 20 shows the number of messages when the network converges in scenario 2 using the modified parameters. We highlight an initial peak of 250 messages at the 20th second, but the rest of time the number of messages is around 45 messages. Compared to the results shown in Figure 17, the average number of messages has decreased around 15 messages.

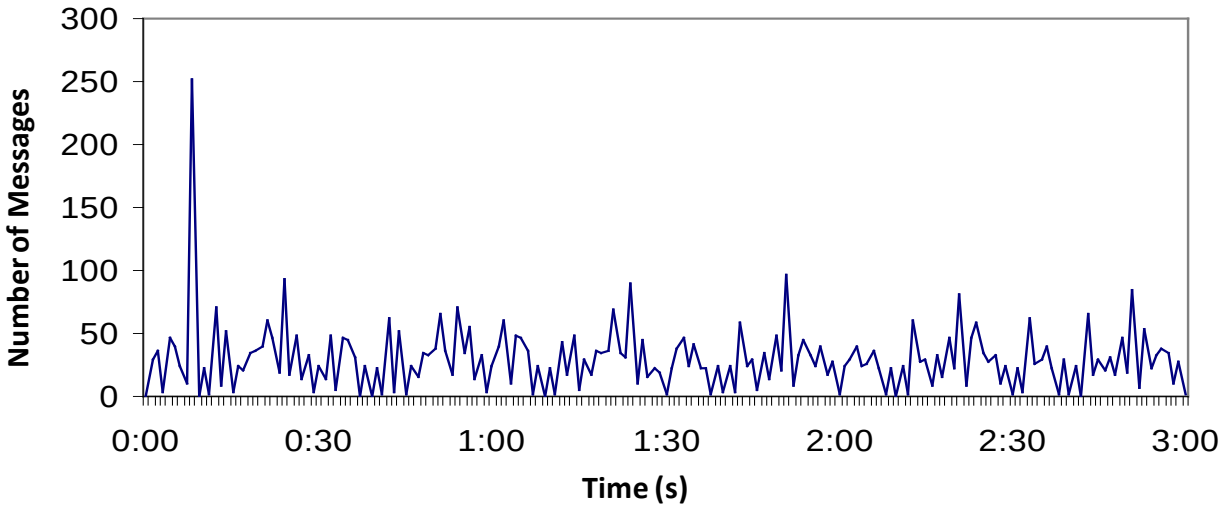

Figure 20. Number of messages observed when network converges in scenario 2, using modified parameters.

Finally, Figure 21 shows the number of broadcast messages observed through the network in scenario 2 using the modified parameters. This case presents the lowest number of broadcast messages. We only highlight 2 peaks registered at 60th second and 139th second with values of 8 and 6 broadcast messages, respectively. The rest of time, we only register sporadic broadcast messages without a defined time pattern.

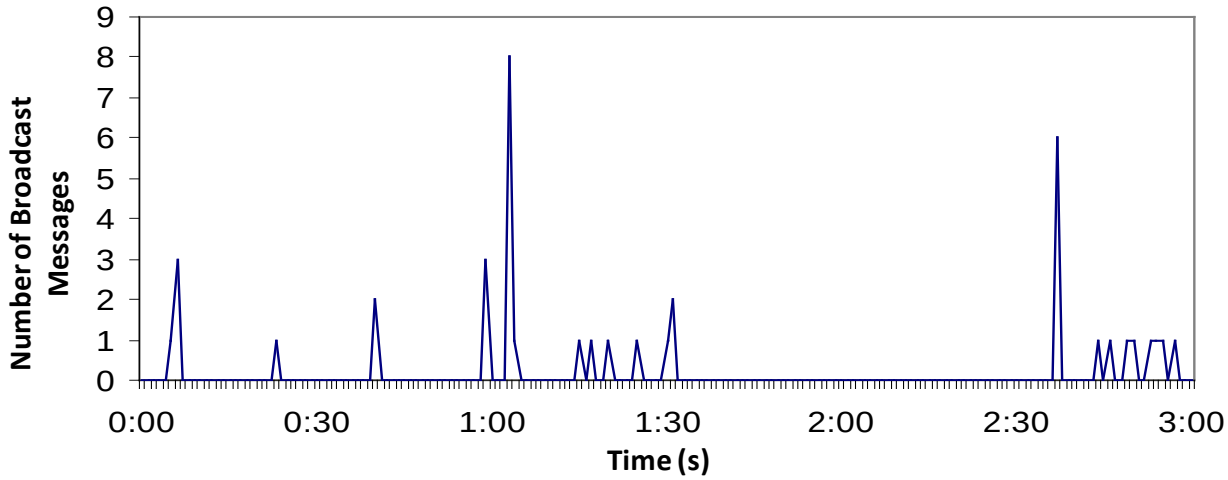

Figure 21. Number of broadcast messages observed when network converges in scenario 2, using modified parameters.

Finally, Table 2 shows the summary of the average values gathered in each test. As we can see, the traffic generated by our algorithm is very low.

	Scenario	**1**	**2**	**1**	**2**
	Parameters used	**By default parameters**		**Modified parameters**	
Output average values	Average value of Bytes/s	2546.98	2836.19	2389.31	2103.87
	Average value of Broadcast/s	0.19	0.29	0.22	0.20
	Average value of messages/s	31.93	35.31	29.56	26.59

Table 2. Summary of results.

As a conclusion of this test, we can say that our protocol present good performance in terms of Bytes/s, Broadcast messages per second and message/s. In addition the best results are obtained in 2nd scenario when modified parameters are used.

7. Conclusion

In this paper we have proposed hierarchic architecture, which includes fault tolerance and security, and permits exchanging data, services, computing and storage resources between all interconnected eHealth/mHealth clouds. The hierarchic architecture is based on two layers: the organization layer, the distribution layer. Organization layer is formed by Org_nodes which are used to establish connections between Org_Nodes and Dist_nodes of different clouds. The routing of messages is based on node's ID. For this reason, the designed frames do not depend on IPv4 addresses and therefore becomes very easy the architecture migration to IPv6 or to any other network layer protocol. We have defined 2 main parameters to establish the suitability of a node. These parameters depend on the node bandwidth (in kbps), the number of available connections, the maximum number of connections and their % of available load, among others. In addition, we have presented a secure system to ensure the veracity of identity and algorithm for the network recovery after removing a node, which is very important in eHealth/mHealth systems.

The test bench results show that after removing a node for security reasons, the network recovers quickly without thereby generates an excessive volume of messages. The bandwidth consumed by our proposal is not too high and proves it viability. We have also seen that increasing the number of eHealth/mHealth clouds, the number of broadcast messages remains approximately constant, so adding more eHealth/mHealth clouds do not overload the system. This is very interesting, when working with devices of low computational capacities, such as mobile devices or smartphones.

Our future work is focused on testing the network performance when other routing protocols are used. We also would like to implement our architecture in open eHealth/mHealth clouds and compare the differences in their operation. We want to adapt our architecture to other kind of networks and technologies, such as mesh networks and spontaneous ad-hoc networks [53] and try to work with other type of data such as multimedia content. Finally, we will study the energy consumption of all the nodes of the network in all these new implementations.

References

[1] The Fifty-eighth World Health Assembly, Resolutions and Decisions. Document: A58/21. Available at: http://www.who.int/healthacademy/media/WHA58-28-en.pdf. [Last access: Dec. 30, 2014]

[2] World Health organization. Topics of eHealth. In WHO website. Available at: http://www.who.int/topics/eHealth/en/. [Last access: Dec. 30, 2014]

[3] Pickup, J. C., Freeman, S. C., and Sutton, A. J. (2011). Glycaemic control in type 1 diabetes during real time continuous glucose monitoring compared with self monitoring of blood glucose: meta-analysis of randomised controlled trials using individual patient data. TheBMJ, 343:d3805.

[4] Promotional material Digital health: working in partnership. Department of Health. UK. (2014) Available at: https://www.gov.uk/government/publications/digital-health-working-in-partnership/digital-health-working-in-partnerships#digital-health---harnessing-technology-for-patient-benefit. [Last access: Dec. 30, 2014]

[5] eHealth for a Healthier Europe!– opportunities for a better use of healthcare resources. Available at: https://joinup.ec.europa.eu/sites/default/files/files_epractice/sites/eHealth%20for%20a%20Healthier%20Europe%20-%20Opportunities%20for%20a%20better%20use%20of%20healthcare%20resources.pdf. [Last access: Dec. 30, 2014]

[6] Adibi, S. (2012). Link technologies and BlackBerry mobile health (mHealth) solutions: a review. IEEE Transactions on Information Technology in Biomedicine, 16(4), 586-597.

[7] Chiarini, G., Ray, P., Akter, S., Masella, C., and Ganz, A. (2013). mHealth technologies for chronic diseases and elders: A systematic review. IEEE Journal on Selected Areas in Communications, 31(9), 6-18.

[8] Lopes, I. M., Silva, B. M., Rodrigues, J. J., Lloret, J., and Proenca, M. L. (2011). A mobile health monitoring solution for weight control. In proceedings of the 2011 International Conference on Wireless Communications and Signal Processing (WCSP 2011). Nanjing,(China), Nov. 9-11, 2011. (pp. 1-5)

[9] Lopes, I.M., Silva, B.M., Rodrigues, J.J.P.C., Lloret, J. (2012) Performance Evaluation of Cooperation Mechanisms for m-Health Applications. In proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM 2012). Anaheim (California-USA), Dec 3-7, 2012.

[10] Kyriacou, E. C., Pattichis, C. S., and Pattichis, M. S. (2009). An overview of recent health care support systems for eEmergency and mHealth applications. In proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009). Hilton Minneapolis, Minnesota, USA. Sept. 2-6, 2009. (pp. 1246-1249).

[11] Nkosi, M. T., and Mekuria, F. (2010). Cloud computing for enhanced mobile health applications. In proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom 2010), Indianapolis, IN, USA, Nov. 30 2010-Dec. 3 2010, (pp. 629-633).

[12] Sultan, N. (2014). Making use of cloud computing for healthcare provision: Opportunities and challenges. International Journal of Information Management,34(2), 177-184.

[13] Pandey, S., Voorsluys, W., Niu, S., Khandoker, A., and Buyya, R. (2012). An autonomic cloud environment for hosting ECG data analysis services. Future Generation Computer Systems, 28(1), 147-154.

[14] Xia, H., Asif, I., and Zhao, X. (2013). Cloud-ECG for real time ECG monitoring and analysis. Computer methods and programs in biomedicine, 110(3):253-259.

[15] Bourouis, A., Feham, M., and Bouchachia, A. (2012). A new architecture of a ubiquitous health monitoring system: a prototype of cloud mobile health monitoring system. arXiv preprint. Reference: arXiv:1205.6910.

[16] Chen, K. R., Lin, Y. L., and Huang, M. S. (2011). A mobile biomedical device by novel antenna technology for cloud computing resource toward pervasive healthcare. In proceedings of the 11th International Conference on Bioinformatics and Bioengineering (BIBE 2011), Oct. 24-26, 2011, Taichung (Taiwan). (pp. 133-136).

[17] Lacuesta, R., Lloret, J., Sendra, S., and Peñalver, L. (2014), Spontaneous Ad Hoc Mobile Cloud Computing Network, Scientific World Journal, 2014(Article ID 232419): 1-19

[18] Ghafoor, K. Z., Bakar, K. A., Mohammed, M. A., and Lloret, J., (2013)Vehicular Cloud Computing: Trends and Challenges (Chapter 14), In Mobile Networks and Cloud computing Convergence for Progressive Services and Applications. IGI Global. Pp. 262-274. DOI: 10.4018/978-1-4666-4781-7.ch014

[19] Wan, J., Zhang, D., Zhao, S., Yang L. T., and Lloret J. (2014), Context-Aware Vehicular Cyber-Physical Systems with Cloud Support: Architecture, Challenges and Solutions, IEEE Communications Magazine, Vol. 52, Issue 8, Pp. 106-113. Doi: 10.1109/MCOM.2014.6871677

[20] Rodrigues, J. J. P. C., Zhou, L., Mendes, L. D. P., Lin, K., and Lloret, J., (2012), Distributed Media-Aware Flow Scheduling in Cloud Computing Environment, Computer Communications. 35(15):1819-1827.

[21] Dutta,R., Annappa, B., (2014), Protection of data in unsecured public cloud environment with open, vulnerable networks using threshold-based secret sharing, Network Protocols and Algorithms, 6(1): 58-75.

[22] Modares, H., Lloret, J., Moravejosharieh,A., Salleh, R. (2013), Security in Mobile Cloud Computing (Chapter 5). In Mobile Networks and Cloud computing Convergence for Progressive Services and Applications. IGI Global. pp. 79-91.

[23] Mehmood, A., Song, H., Lloret, J., (2014), Multi-Agent based Framework for Secure and Reliable Communication among Open Clouds, Network Protocols and Algorithms, 6(4):60-76.

[24] Mendes, L. D. P., Rodrigues, J. J. P. C., Lloret, J. and Sendra, S., (2014). Cross-layer Dynamic Admission Control for Cloud-based Multimedia Sensor Networks. IEEE Systems Journal. 8(1): 235 – 246.

[25] Xiong, J., Li, F., Ma, J., Liu, X., Yao, Z., and Chen, P. S. (2014). A full lifecycle privacy protection scheme for sensitive data in cloud computing. Peer-to-Peer Networking and Applications, Vol: June 2014, 1-13.

[26] Yang, H., Kim, H., and Mtonga, K. (2014). An efficient privacy-preserving authentication scheme with adaptive key evolution in remote health monitoring system. Peer-to-Peer Networking and Applications, Vol: June 2014, 1-11.

[27] Silva, B. M., Rodrigues, J. J., Canelo, F., Lopes, I. M., and Lloret, J. (2014). Towards a cooperative security system for mobile-health applications. Electronic Commerce Research, Vol: November 2014, 1-27.

[28] Flynn, D., Gregory, P., Makki, H., and Gabbay, M. (2009). Expectations and experiences of eHealth in primary care: a qualitative practice-based investigation. International journal of medical informatics, 78(9), 588-604.

[29] Thampi, S. M. (2010), Survey of Search and Replication Schemes in Unstructured P2P Networks, Network Protocols and Algorithms, 2(1): 93-131

[30] Khan, S. M., Mallesh, N., Nambiar, A., and Wright,M., (2010). The Dynamics of Salsa: A Robust Structured P2P System, Network Protocols and Algorithms, 2(4): 40-60.

[31] Garcia, M., Hammoumi, M., Canovas, A., and Lloret, J. (2011). Controlling P2P File-Sharing Networks' Traffic. Network Protocols and Algorithms, 3(4), 54-92.

[32] Lloret, J., Garcia, M., Tomas, J., Rodrigues, J. J. P. C., (2014). Architecture and Protocol for InterCloud Communication, Information Sciences, 258(-): 434-451.

[33] Chowdhury, C. R., A Survey of Cloud Based Health Care System. (2014). International Journal of Innovative Research in Computer and Communication Engineering. 2(8):5477-5481.

[34] Ghosh, R., Papapanagiotou, I., Boloor, K. A. (2014) Survey on Research Initiatives for Healthcare Clouds. Cloud Computing Applications for Quality Health Care Delivery. IGI Global, 2014, pp. 1-18.

[35] Donahue, S., (2010) Can Cloud Computing Help Fix Health Care?. Cloudbook Journal , 1(6): 1-6.

[36] Deng, M., Petkovic, M., Nalin, M., Baroni, I.,A., (2011) Home Healthcare System in the Cloud--Addressing Security and Privacy Challenges. In proceedings of the 2011 IEEE International Conference on Cloud Computing (CLOUD 2011), Jul. 4-9, 2011, Washington, DC, USA, (pp. 549-556).

[37] Sultan, N., (2014) Making use of cloud computing for healthcare provision: Opportunities and challenges. International Journal of Information Management, 34(2):177-184.

[38] Nkosi, M. T., Mekuria, F. (2010) Cloud computing for enhanced mobile health applications. In proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom 2010), Nov.30-Dec.3, 2010. Indianapolis USA.(pp. 629-633).

[39] Wang, X., Gui, Q., Liu, B., Chen, Y., and Jin, Z., (2013) Leveraging mobile cloud for telemedicine: a performance study in medical monitoring. In proceedings of the 39th Annual Northeast Bioengineering Conference (NEBEC 2013). Syracuse, NY, USA, Apr. 5-7, 2013. (pp. 49-50).

[40] Alamri, A., (2012) Cloud-based e-health multimedia framework for heterogeneous network. In proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW 2012), Melbourne, Australia, Jul. 9-13, 2012. (pp. 447-452).

[41] Constantinescu, L., Kim, J., Feng, D. D. (2012). Sparkmed: A framework for dynamic integration of multimedia medical data into distributed m-health systems. Information Technology in Biomedicine, IEEE Transactions on, 16(1): 40-52.

[42] Botts, N., Thoms, B., Noamani, A., Horan, T. A., (2010). Cloud computing architectures for the underserved: Public health cyberinfrastructures through a network of healthatms. In proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS 2010), Jan. 5-8, 2010, Honolulu, HI, USA.(pp. 1-10).

[43] Fan, L., Buchanan, W., Thummler, C., Lo, O., Khedim, A., Uthmani, O., Lawson, A., Bell, D., (2011). DACAR platform for eHealth services cloud. In proceedings of the 2011 IEEE International Conference on Cloud Computing (CLOUD 2011),. Washington, DC, USA, Jul. 4-9, 2011. pp. 219-226.

[44] Ruiz-Zafra, A., Benghazi, K., Noguera, M., Garrido, J. L., (2013), Zappa: An Open Mobile Platform to Build Cloud-Based m-Health Systems, in proceedings of the 4th International Symposium on Ambient Intelligence (ISAmI 2013), Salamanca, (Spain), May 22-24, 2013. (pp 87-94).

[45] Nijon, S., Dickerson, R. F., Asare, P., Li, Q., Hong, D., Stankovic, J. A., Hu, P., Shen, G., Jiang, X., Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM, Taipei, Taiwan, Jun. 25-28, 2013. (pp. 403-416).

[46] Xia, H., Asif, I., Zhao, X., (2013) Cloud-ECG for real time ECG monitoring and analysis. Computer Methods and Programs in Biomedicine. 110(3):253-259.

[47] Lloret, J., Diaz, J.R., Boronat, F., and Jiménez, J.M., (2006), A Fault-Tolerant P2P-based Protocol for Logical Networks Interconnection, In proceedings of the International Conference on Networking and Services (ICNS'06), July 16-22, 2006. Silicon Valley (USA),

[48] Lloret, J., Palau, C., Boronat, F., Tomas, J. (2008), Improving networks using group-based topologies, Computer Communications 31(14):3438-3450.

[49] Lloret, J., Boronat Segui, F., Palau, C., and Esteve, M. (2005), "Two Levels SPF-Based System to Interconnect Partially Decentralized P2P File Sharing Networks", In proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services.(ICAS-ICNS 2005). Oct. 23-28.2005.Papeete, Tahiti. (pp. 39-39)

[50] Cramer, C., Kutzner, K., and Fuhrmann, T. (2004). Bootstrapping locality-aware P2P networks. In proceedings of the 12th IEEE International Conference on Networks (ICON 2004). Nov. 16 - 19, 2004 Singapore (pp. 357-361).

[51] FIPS 180-1 - Secure Hash Standard, SHA-1. National Institute of Standards and Technology. http://www.itl.nist.gov/fipspubs/fip180-1.htm [Last access: Dec. 30, 2014]

[52] Eastlake D., Jones P., US Secure Hash Algorithm 1 (SHA1),(2001). In IETF website, Available at: http://www.ietf.org/rfc/rfc3174.txt [Last access: March 20, 2015]

[53] Lacuesta, R., Lloret, J., Garcia, M., and Peñalver, L. (2011), Two Secure and Energy-Saving Spontaneous Ad-Hoc Protocol for Wireless Mesh Client networks, Journal of Network and Computer Applications, 3(2): 492-505.