XMPP Distributed Topology as a Potential Solution for Fog Computing

Jonathan Bar-Magen Numhauser, Jose Antonio Gutierrez de Mesa
Department of Computer Science
University of Alcala
Alcala de Henares, Spain
e-mail: {jonathan.bar-magen, jantonio.gutierrez}@uah.es

***Abstract*—Fog Computing is potentially harmful towards existing Cloud Computing systems as well as Big Data structures. Lack of privacy and potentially unauthorized content distribution are some of many issues to solve. As a possible solution, we propose in the following paper the definition of a new Network Topology as well as a working methodology to reduce its impact. It serves as an introduction to our investigation line and for future works that aim to solve the effect of the Fog Computing.**

Keywords-Fog Computing; topology; network; Big Data; XMPP.

I. INTRODUCTION

Our previous studies into Fog Computing [1][2] were basically introductory; being such an innovative subject devised by our research team, we defined its identity, and eventually obtained a structure that established what Fog Computing is, as well as its implications on Big Data structures. In this paper, we are following the established research steps, and present later the second part of the research process, which is equivalent to the first technical phase of the research guideline.

Fog Computing can be perceived both in large Cloud systems and Big Data structures, making reference to the growing difficulties in accessing objectively to information. These results in a lack of quality of the obtained content and, in some cases, even it's dumping. The effects of Fog Computing on Cloud Computing and Big Data systems may vary; yet, a common aspect that can be extracted are the limitations in accurate content distribution, an issue that for long has been tackled with the creation of metrics that attempt to improve such [9].

The influence of a variety of entities in the cloud/big data market impacts the magnitude of Fog Computing [6], as described by Agrawal et al. [4]. For this study, we supposed a network structure of Cloud or Big Data where we may find and advanced stage for Fog Computing. To bring a solution to such advanced stage, we are about to propose a number of methods, which were later tested. These results are examined for optimization and future research steps, as well as feedback for collaborative methodologies [1]. The first section of this paper will expose the necessary requirements needed for Fog Computing to appear in Cloud Computing and Big Data. It will be followed by a description of the evolution of Fog Computing in such structures. Once we establish the necessary requirements for a Big Data structure affected by Fog, this paper will explain the first steps taken to find a solution to such a situation, and show the development steps taken to find possible alternatives to solve the existence of Fog.

This paper will start by describing how Fog Computing manifests itself in Big Data structures; it will be followed by the analysis of algorithms to manage information systems. In the Methodology and Technological Background sections that follow, a study of the methods to use in the research work as well as the technological necessities will be treated. Once established the background variables, in the Investigation process section, the steps taken in the overall research work will be described to details.

After the research section, the Development section will establish the development process that took part in our work. Finally, the conclusion section will summarize all the work that took place in this project phase.

It is imperative to highlight that this is the first phase of a three-phase research process that will culminate in a complete empiric result analysis, which implies that in the early phases no more than topological and structural results will be exposed.

II. FOG COMPUTING IN BIG DATA

Big Data structures in the last few years have been acquiring a central role [5]. As the following evolution step of Cloud Computing, before entering into it, this evolution has to take place with a number of precautions to ensure that the Fog do not pass towards the Big Data structures [3].

Cloud Computing and Big Data function on the topology of third party storage information [4]. This structure advocates that each node of the network relies its information on a centralized data bank and allows the powerful core mainframes to process such information and obtain richer results. In case of Cloud Computing, the second part with regard to processing is not as evolved as in Big Data, in which the core calculations are a fundamental part of the system [5].

When creating these types of data structures, we have to consider the following issues: Relying the impact of size and memory in the mainframe, and Hardware costs that requires an effective and optimized algorithm to manage such information.

The research lines of semantic and indexation are having a major importance in many universities and laboratories,

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-299-8

aiming at obtaining better search algorithms and reduce the impact of size and memory in the mainframe. This field of study is of such importance that most of the largest technological entities have independent research departments only to find new and improved search methods.

The need to find such solutions is intertwined with the constant change in the global network topology. The information-based structure is pushing towards studies to dedicate an increasing amount of time and energy in finding metrics and algorithms to improve search engines, and by default the management of information [7].

It is this increase of information structures that we consider is the trigger to the Fog Computing, and so the establishment of our research line [2]. Cloud Computing and Big Data implies a significant amount of information management, information that can be easily manipulated and filtered as it was proven in our previous introductory paper[2]. Fog Computing is the situation in which mass information structures loose control of the attempted impartial nature of information search. If we search for information of type “A”, a system affected by Fog Computing may intentionally or unintentionally filter the resulting information, which will have the effect of a limited visual spectrum for the user.

The causes of this manipulation can be either intentional or non intentional, as a result of the evolution of algorithms and their attempts to achieve perfection in result accuracy.

Fog Computing can be instigated by a number of external variables, those variable that covers numerous fields including security, commerce and politics, all of which have the same objective: the filtering or obscurity of information of type “A” behind information of type “B”.

These requirements can be found in most existing Cloud and Big Data networks, commercial and political in search engine information as seen in Bar-Magen [2]. From search engines, like Google Search, and Bing to profile-based networks, like Facebook or Twitter, Fog Computing can be found in each, represented by sponsored manipulated results, lack of user privacy and remote manipulation of sensitive data.

F Fog Computing has to be considered at the moment of designing a Big Data system; in the following sessions, we will expose a possible solution based in topological structure to reduce the impact of Fog in the network systems. It is important to state that Fog Computing cannot be totally solved, but we work to reduce its impact as much as possible.

III. ALGORITHMS AND INFORMATION MANAGEMENT

A. *Algorithms for result optimization*

Before heading on to the topology analysis and how to use it to solve Fog Computing, we would like to state a number of issues regarding the situation of present algorithms to manage information.

Algorithms to manage information are being devised constantly to improve the visualization of information. The evolution of these algorithms aims to optimize search results on regional and profile learning interests. These algorithms make use of a complex profile map, based on user interests, of the user who searches for such information [6]. The evolution of those algorithms aim to increase the search spectrum to use linked profiles of users related to the alpha user, which today attempts to use the power of profiled networks, like Facebook, and search engines, like Bing. The combination of these two tools attempt to keep optimizing the information management algorithms.

The overall objective of information management algorithms is to achieve a precise result and adapt it to the user’s instant necessities. By now, as a result of our previous studies, we obtained that such attempts to improve search algorithms result in the increment of Fog Computing in the information networks [2]. Many of the current top content management entities use such algorithms to stay at the highest position in search results success hits, pushing forward the algorithm evolution. Yet, this evolution ends up in the unconscious filtering of a larger amount of information, prioritizing one type of information over the other.

B. *Information management and increase revenue*

Information management had a main role in advanced civilizations and its motives are so vast that it merits a study of its own. In our case, the use of several algorithms in information management allows the benefit of several entities above others [8]. Those algorithms are mold to deal with the specific variables; the variables are, in general, established by entities that benefit from information management engines, which eventually render them partial and biased.

As a result of such partiality, we decided to approach the definition of a topological structure for network components, and once the structure is established define a new brand of algorithms that will manage the new network structure. Some of those resulting algorithms that will be exposed in the following sections are based on standardized XMPP/XEP XML protocols as well as localized environmental frameworks; for example, the iOS XMPPFramework created by Hanson [19].

The information management in a new topology will result in a different concept of data distribution and control. The most significant difference will be noticed in the content distribution and recollection, reducing the centralized management of such. This may be noticed in the choice of hardware for our project, being strongly based on mobile devices, and considering the dynamic characteristic of those devices, a swarm based structural functionality may be adopted imitating Swarm-Computing structures, which will imply the creation of a mobility-swarm based architecture [17]. We have chosen such distribution because it will allow us to establish a decentralized smart based peer to peer network in which each node will form part of a complete swarm structure. New content storage and distribution will allow an alternative option to achieve an impartial content exposure, and reduce substantially the Fog in many actual centralized network structures.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-299-8

IV. METHODOLOGY CHOICE

The following step will be to establish the methodology to use in the following research work. To accurately predict the future steps of our work process, it is imperative that an appropriate methodology be chosen. Looking at our previous investigation into collaborative methodologies [1], that contemplated the creation of standardized rules for correct functionality in collaborative work, and alternatively analyze the results of our research in Native Based Development that included the definition of working steps to follow facing a project that spreads a large spectrum of technological environments, we concluded that a combined working technique of both fields is needed for our current project.

Collaborative Methodology requires the participation of a number of entities at the development phase of a project, and so we were able to obtain the collaboration of at least a couple of entities to work on this development. A second issue in Collaborative Methodology work is the dependency on collaborative tools, and the way to manage them in the course of the project development. The tools are related to the agile nature of this methodology, translated to the creation of a number of roles dedicated to the completion of certain objectives.

For this project, we adopted a Collaborative Methodology because we are also familiar with its workflow and phases allowing us to easily adapt the objectives of our software and hardware development.

As a result of the technical specification, it will be necessary to develop on each device, natively the intelligence necessary to manage the content transaction between each device. Combining it with Collaborative Methodology it will substantially reduce the work load, ensuring the coordination and synchronization of each member and each component being developed.

The resulting endeavor was influenced by the chosen methodology and allowed us to achieve a number of results that were not initially expected in the hypothesis.

V. TECHNOLOGICAL BACKGROUND

Defining the correct background variables for our research process will help defining the basic development necessities to comply with our objectives. Peer-to-peer, Swarm Computing and privacy and security measures will be some of the issues that were studied as a background analysis.

A. Peer-to-Peer Concept

The studies on peer-to-peer have been numerous in the last years [14][15][16]. In our work we decided to make use of this network structure to attempt a possible by pass of centralized information flow, and so, create a new data structure for transactions that may permit the establishment of a standardized information distribution. This new data structure will be strongly based on peer-to-peer topological distribution used for our practice XMPP/Jabber system.

This new information distribution standard will be part of the main study to solve Fog. What we aim to describe in this section is that the concept of peer-to-peer information transaction has been adopted as an initial background requirement to solve the issue at hand.

Peer-to-peer network structures can be widely seen in a number of file transaction systems, as well as communication clients, in particular, based on either Adobe Cirrus multimedia server, or, in our case, the XMPP Jabber server [14].

XMPP server supports partial peer-to-peer connection in the messaging method, and full peer-to-peer transactions with the use of the Jingle extension [18]. With the available channels for distribution of information, we will be able to establish a main working method to follow in the research process.

B. Swarm Computing

As established in a number of places in the previous section, the adoption of Swarm Computing development rules will allow us to emulate a Swarm Computing behavior in the overall peer-to-peer system that will act as the solution to the Fog Computing issue.

Swarm Computing [17] refers to the technological field of independent processing modules coordinating to offer unified and larger information in a certain spectrum. Swarm Computing is strongly linked to the described in the previous point, for our research peer-to-peer and Swarm Computing refers to the same network structure. Depending on its use, the network structure can be more or less dependent on a centralized source.

In our study case, we opt to define the structure as mostly decentralized, offering the minimum required common points for successful communication establishment. An analysis of Swarm Computing on the propagation of Fog Computing and its synchronization with the rest of the background variables will follow.

C. Privacy and Security

Privacy and security of the shared data is a serious issue to be concerned with regarding Big Data structures. At the present, high security risks that can be found in common Big Data and Cloud Computing structures drive users to lower their trust levels in those systems, thus reducing the efficiency of Big Data systems.

Big Data thrives on the possible combination of information with the purpose of obtaining new and more optimal content. A great example is disease spreading, that with an efficient Big Data structure, can be better controlled and offer better solutions.

When we regard the issue of security and privacy in Big Data, we should always consider two main sources of difficulties, either the information exploits by the central system, which is the existence of an impartial and abusive central administration, or exploits driven by individual attackers, which can result in the trafficking of information by third parties.

Both issues form part of Fog Computing, and that is why we are driven in our research to find a trusted solution to these problems.

We believe that through the establishment of a new network structure, as well as the use of a number of studied technologies, we may be able to increase the security and privacy of members in a Big Data structure, and so reduce the impact of the Fog Computing in the system.

VI. Research Process

To initiate the first phase of our research, once defined the background variables, we proceeded with the establishment of the network structure that will form the new communication systems.

We seek to achieve a new method of communication and information distribution that may emulate a Big Data or Cloud Computing structure and offer a counter solution incase of a centralized structure affected by Fog Computing. Fogs in large information structures are the main issue to solve, and we started by defining a peer-to-peer based Swarm Computing topological system.

The system will require the functionality of XMPP/Jabber server to allow a centralized address access, serving as a directory for the members of the Big Data structure to communicate. To solve the massive information filtering systems, we studied the importance of background interaction of services with the technological environment. Each member of the swarm will be considered as a component from now on, with an associated entity id.

A. Component's background services

In our project we defined that each component of the swarm should contain a minimum number of services to function autonomously, considering it as part of its intelligence. This behavior will be achieved by using appropriate hardware and software characteristics. The basic hardware/software will be an operating system with I/O capabilities and network connection. In this specific test case we made use of an iOS device, and we categorized it as a component in the swarm.

Background services are what allowed components of the swarm to react to environmental changes, defining the importance of these services, that without them all the theory of our topological structure will fail. In the first phase, we opted to develop simple software to simulate the swarm activity in a component, and see the behavior of such component on a macro scale.

Those services will constantly communicate with other services in other components, and so complete the swarm behavior. It is through those services that we will achieve a complex Big Data counterpart to the traditional centralized structure.

B. Native-Based Development for Components

Even though we used for this sample case an iOS based environment, we should consider the importance of a native based development in our research process. Each component will be developed according to its environmental variables, on a software level. These native environments will cover iOS, Android, or AS3 in case of a PC device.

Their common denominator will be the communication protocol, which will be the Jingle XMPP protocol. Their communication will pass through a common server that will route the connection between two or more members of the swarm. That is why the use of a proper native development methodology is essential for a successful result.

C. Component's Characteristics.

If we consider each component to have enough autonomy to send and receive information, as well as to store it, we will notice the a creation of a peer-to-peer structure that at the same time also simulates a distributed network system and a Big Data structure.

By increasing the amount of sensors and interaction abilities of the components with its environment, we are successfully creating a richer swarm and an optimal information distribution algorithm. The importance of one type of information in regard to another will be defined not by centralized entities algorithms or interests, but by the same components that form the swarm.

The propagation of information in the swarm will exclusively depend on the components that form part of the swarm. If we compare it with the existing content distribution systems that make use of components as extremities instead of independent members of a swarm, we will notice that the need to route by content packages through a centralized device will result in a non arbitrary control of the obtained content, which will eventually increase the Fog in the system.

D. Lighthouse concept

In our new structure, we attempt to define each component in the swarm as independent enough to behave as a lighthouse in regard to information distribution. Components will store information, and propagate it as long as it will see fit, depending on the surrounding swarm members. Such is the dependency that eventually there may be a case in which information will not transcend more than one component or node.

This structure increases the efficiency of security and privacy of user information in the overall network, allowing him or her to freely hide and delete information from the general swarm. The lighthouse effect in the swarm structure will be the core solution for Fog Computing, allowing the component to decide either autonomously or by manual intervention what information should or should not be propagated.

This will instigate the need to establish new content treatment algorithms, and so we will introduce an initial phase for content manipulation in the swarm.

E. Content mangement in the swarm

Algorithms to manage content are vast in present systems, yet if we pretend to create a secure and private content control structure, as well as a trusted and optimized access to content, it is essential to establish a structure that will permit the evolution of content management and its algorithms.

Such changes should be open for either autonomous improvement as well as manual intervention. And it will have to contemplate the possible random improvement of information management algorithms. All this requisites will be treated towards the second phase of our study. Nonetheless we will establish a basic communication standard, on the foundation of XMPP XEP extensions, as well as the creation of new extensions [15]. The algorithms will be improved with time and increase in efficiency as the project advances. It will also contemplate the use of smart components as well as intermediate nodes that will allow a better flow of the content, but not its manipulation, so we can avoid a possible appearance of Fog in the system.

The increase in components complexity can either increase the efficiency of the structure or decrease it, depending on the algorithmic methods that are being followed.

F. The Work Process

Once we obtained all the pre requisites from our investigation into peer-to-peer, Swarm Computing, content management algorithms and XMPP servers, we proceeded with the creation of our first swarm based network, and analyze the possible existence of variables that may result in the appearance of Fog Computing.

VII. DEVELOPMENT PROCESS

At first we devised a network structure that will comply with the aspects described in the previous sections. The use of complex swarm components will suppose on one hand a higher cost in development, but on the other hand it may result in a greater source of information resulting from the created algorithms.

Initially we implemented a simple communication program that will use the network structure to simulate a peer-to-peer communication. The adaptation of XMPP libraries to the development environments was a costly process, but once it was completed, the propagation of the same enterprise to different native environments was successfully optimized.

The creation of custom extensions required a previous study of the limitation that XMPP structures have, in particular de XEP protocols, and how they restrict the exchange of content. The extensions required to be registered on a public domain. XMPP uses XML language structure, which also needed the preparation of modules to interpret those extensions in each network component or device.

Once established the peer-to-peer communication structure, we run some tests on the overall process of content transaction obtaining favorable results in regard to latency and bidirectional interaction. The bidirectional communication resulted in a fast and almost instantaneous interaction, having used in our case components existing in the Madrid surrounding area with a small component number of 20. Later on we passed the test components to a larger distance, from Latin America to Europe, resulting in favorable and low latency results. An average instantaneous message inside the Madrid area took 35ms to be received from source to destination. On the other hand, a message from Spain to Brazil took longer but not on a level out of our prediction: 125ms.

On pure peer-to-peer communication through Jingle node, latency was calculated using Wireshark package reader, receiving a number of interesting results. Latency inside the Madrid area was under 50ms, while latency between Madrid and Sao Paulo was around 230ms.

The following step was to improve the communication tools, and allow a higher quality of content exchange. Implementing a Jingle extension could do this. For this process we made use of a variety of sources, which in their core ended up resulting in the use of the Jingle Library supported by the Google Development Group.

This library created in C, is easily adapted to a variety of environments allowing a smoother process of integration in our swarm components. The result was favorable, and we were able to transmit audio content from one component to another using the peer-to-peer communication structure. Latency results on this case depended on a number of variables, including the unified size of the byte stream being sent, but in an overall analysis we obtained that the connection speed between the two nodes was favorable as well.

On regard to the internal behavior of the components, we established a number of methods to store relevant information. Initially we allowed the user to define if the swarm component will be a storage node and a propagator, or only a propagator. The difference will be noticed on the level of priority the component will have, where the storage nodes are of higher security priority. Storage nodes not only send their information, but also store the information for future distribution, and in case of deletion, if such information is not replicated in another storage node marked as origin, it will be completely removed from the system.

The available table management environment in the mobile devices, through the use of SQL structures, allowed the creation of an improved communication flow. By having a strong intelligence base, each node of the swarm can work as storage and propagator, as well as contain a variety of complex algorithms.

The final step of our first research phase was the creation of a unified extension to communicate between swarm components, and as a result analyze the frequency in which background services used to exploit these extensions. We noticed that some simple commands could be avoided from being used through Jingle extensions. Jingle uses an open socket channel between two nodes, and such channels may require a greater management level from the components software, which will result in higher complexity. Push commands could be passed by a normal extension and an IQ or Message type, which resulted in a cleaner and less dependent communication.

Information requests could also be executed through push requests instead of Jingle, but the greatest inconvenient will be in the limitation of content transaction and content size.

The background services that were implemented, in this case normal message reception as well as automatic light level management, worked according to our expectations. Later on we also incorporated the detection of the components movement based on the accelerometer. All those sensors sent information to the components that requested it, and even in some cases activated other components on the execution of a certain event, imitating the behavior of a smart swarm system.

The best example was the execution of an alert on the main control panel in the PC version of the program when the device fell from a high altitude. Thus it confirms the possible interaction of swarm elements, and the potential improvements for more complex content exchange algorithms.

VIII. Conclusion

For this research project we established a number of objectives to achieve. The first was to obtain the necessary requirements to identify a big data structure affected by Fog Computing. This was detailed in the first sections of this paper. The following goal was to obtain a study of the appropriate methodologies for the development of this project and the creation of this new content structure [2].

When we look into the topological study of a network structure, inflicted by Fog Computing, we have to consider privacy, security, and scalability and efficiency issues. These were our considerations when we approached the second step of our project. Basing on our experience from previous studies we concluded that opting to use a Collaborative Methodology, and a native based development workflow, would be the best choice confronting the existence of a swarm peer-to-peer network structure.

We should point that those methodologies will be composed of agile elements such as SCRUM amongst others. When we proceeded to the practical phase, we encountered the need to establish the technological pre requisites, as well as the overall structure that our network should be based on, as is shown by the compared network structures in Figure 1. From this study we devised the use of a swarm structure, and its combination with a decentralized peer-to-peer communication standard [16].

As part of our study we also decided to use XMPP protocol to achieve a true peer-to-peer communication structure, and also the use of smart components that will are parts of the swarm network [18].

The development ended up in the creation of a software, that can be run in a variety of components that allowed us to achieve substantial empiric results that are being studied to be exposed in the second phase of our research, and that in summary, show the possible peer-to-peer communication between components that are part of a decentralized swarm network, and even the possible incorporation of independent background services that will add complexity to the content manipulation.

Being this the first phase of a three-step project, we cannot yet inform on empiric results of the impact of this system on Fog Computing. What we can report is that a first phase of communication in a new swarm network structure was possible, and that the Fog Computing was reduced by allowing information to be stored in the components and so creating an alternative for improvements in privacy and security.

Future study will bring us more empiric results that may be compared to existing centralized structures, being one of them a Z Mainframe execution of such a network structure.

References

[1] J. Bar-Magen, A. Garcia-Cabot, E. Garcia, L. de-Marcos, and JA. Gutierrez de Mesa. “Collaborative Network Development for an Embedded Framework”. In: Uden L, Herrera F, Pérez JB, Corchado Rodríguez JM, editors. 7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing: Springer Berlin Heidelberg; 2013. p. 443-453.

[2] J. Bar-Magen, “Fog Computing- Introduction to a new Cloud evolution”. In: Jose F. Fornies Casals, Paulina Numhauser, editor. Escrituras Silenciadas: El paisaje como Historiografia. 1st ed. Alcala de Henares, Madrid, Spain: UAH; 2013. p. 111-126.

[3] F. Frankel and R. Reid,Big data: “Distilling meaning from data.” Nature 2008 09/04;455(7209):30-30.

[4] D. Agrawal, S. Das, and A. El Abbadi. “Big data and cloud computing: current state and future opportunities.” In *Proceedings of the 14th International Conference on Extending Database Technology* (EDBT/ICDT '11), Anastasia Ailamaki, Sihem Amer-Yahia, Jignesh Pate, Tore Risch, Pierre Senellart, and Julia Stoyanovich (Eds.). ACM, New York, NY, USA, 2011, pp. 530-533.

[5] C. Lynch, “Big data: How do your data grow?” Nature 2008 09/04;455(7209): pp. 28-29.

[6] J. Brandon, “Living in the Cloud.” PC Magazine 2008;27(8): pp.19-20.

[7] D. Angeli, “A cost-effective cloud computing framework for accelerating multimedia communication simulations.” Journal of Parallel & Distributed Computing 2012;72(10): pp.1373-1385.

[8] MR. Nelson, “The Cloud, the Crowd, and Public Policy.” Issues in Science & Technology 2009;25(4): pp.71-76.

[9] S.J. Stolfo, M.B Salem., and A.D. Keromytis, "Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud," *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on* , vol., no., pp.125,128, 24-25 May 2012

[10] A.S. Yüksel, M.E. Yuksel, and A.H. Zaim, "An Approach for Protecting Privacy on Social Networks," *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on* , vol., no., pp.154,159, 22-27 Aug. 2010

[11] R. Buyya, J. Broberg, and A.M. Goscinski. 2011. “*Cloud Computing Principles and Paradigms*.” Wiley Publishing.

[12] K. McDonald, “Above the Clouds: Managing Risk in the World of Cloud Computing.” 2010.

[13] E. Boritz and W. Gyun No. 2009. “A Gap in Perceived Importance of Privacy Policies between Individuals and Companies.” In *Proceedings of the 2009 World Congress on Privacy, Security, Trust and the Management of e-Business* (CONGRESS '09). IEEE Computer Society, Washington, DC, USA, 181-192

[14] DA Bryan, BB Lowekamp, and C. Jennings, "SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System." Advanced Architectures and Algorithms for Internet Delivery and Applications, 2005 AAA-IDEA 2005 First International Workshop on 2005: pp. 42-49.

[15] C. dos Santos, S. Cechin, L. Granville, M. Almeida, and L. Tarouco, "On the performance of employing presence services in P2P-based network management systems." In *Proceedings of the 2008 ACM symposium on Applied computing* (SAC '08). ACM, New York, NY, USA, 2090-2094..

[16] L. Stout, M. Murphy, and S. Goasguen," "Kestrel: an XMPP-based framework for many task computing applications." In *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers* (MTAGS '09). ACM, New York, NY, USA, , Article 11 , 6 pages.

[17] G. Beni, "From Swarm Intelligence to Swarm Robotics" Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, 1-9 pages.

[18] P. Saint-Andre, "Jingle: Jabber Does Multimedia," *MultiMedia, IEEE* , vol.14, no.1, pp.90,94, Jan.-March 2007.

[19] R. Hanson, XMPPFrameWork for iOS environments, https://github.com/robbiehanson/XMPPFramework/wiki/IntroToFramework

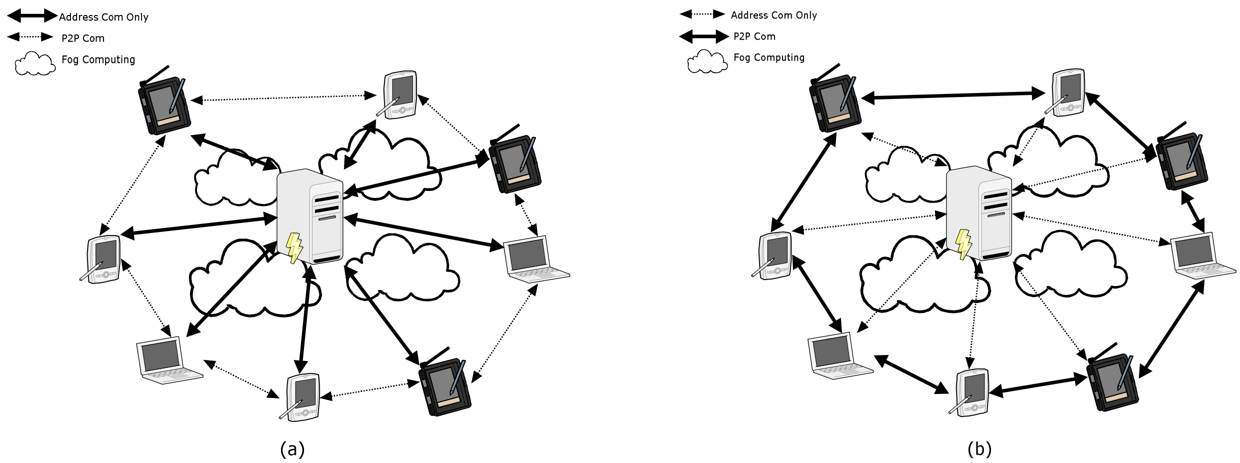

Figure 1. Network structures: (a) Centralized information network system, (b) XMPP P2P based information network system