Efficient Distribution of Location Information and Locating Radio Devices for Dynamic Spectrum Allocation Systems

Toan Hoang*, Josef Noll† and Torleiv Maseng*
Email: toan@unik.no, josef@unik.no and torleiv.maseng@ffi.no

*Norwegian Defense Research Establishment (FFI)
†UNIK - University Graduate Center

***Abstract*—In the recent years, we have seen a clear trend in consumer devices being equipped with multiple radio interfaces such as GSM, UMTS, WiFi. These types of devices will soon be ubiquitous. Since radio frequency is a limited resource, the need for sharing and re-using a radio spectrum will be even more noticeable than before. Studies show that collaboration when allocating a radio spectrum will result in better network performance. This paper proposes a solution that identifies radio devices in a specific area. Starting from the identification of the location of possible interfering devices, better decisions can be made, prior and under spectrum allocation through cooperation. Our solution provides the location information in an overlay network without need for additional exchange of information messages. By storing the geographic location directly in the node-id, all members of the overlay network can extract the exact location of all known nodes. Using geocast, we show that the information provided can be used to optimize existing geocast mechanisms to identify devices. The result is an overlay-based enhanced spectrum allocation system, ObeSA.**

***Index Terms*—Dynamic Spectrum Access, geo-cast, Distance Flooding Protocol, overlay network, ObeSA**

I. Introduction

Cognitive Radio (CR) was first introduced by Mitola in 2000 [1]. Since then, the CR research field has shown great advancements. Evidence of such progression and acceptance was when the FCC approved unlicensed CR devices to access the free frequencies in the TV bands [2]. These frequencies are known as the 'TV white spaces' spectrum. The most important requirement for these radio devices is that they do not cause harmful interference to the primary users (TV receivers).

Studies [3] show that collaboration between radio units result in better network performance. In the field of CR, there have been many studies in physical spectrum sensing. This function is essential for CRs to be able to operate autonomously. Even though physical sensing has seen many advances lately, without coordination and cooperation between CRs, old issues like the "hidden-node" problems are still present. The FFC's report [2] concluded that with the technology available at the time the report was written, it is not possible to have reliable sensing.

We believe that the opening of the 'TV white spaces' is only one step towards a new future where all spectrum are allocated dynamically. In the future, there will be no such thing as primary or secondary users. All radio devices are equal users of the radio frequencies. Radio frequency will become a product that the end-users only lease when needed. Compared to how CR network is defined, for easier understanding, we can imagine that the owners of the frequencies are the primary users and all other radio devices are secondary users. Our vision of the future of spectrum allocation is presented in [4]. This paper is targeted at this type of network.

The focus of this paper is to enhance spectrum allocating by creating a solution for identifying possible interfering devices. In a heterogeneous network, we see a need of a common, dynamic and efficient way to locate devices of interest. By focusing on a solution between the network and application layer, we believe that it will be easier to implement in the different radio devices. With radio devices that have different specifications, such as battery or calculation capacities, we have focused on a solution that has low impact on the devices. By creating an overlay network where geographic position of the radio devices is provided and using a tailored geocast mechanism, we have created a platform which enables radio devices to allocate other devices of interest, which will result in better spectrum allocation through cooperation. The work in this paper has been based on achieving three goals:

- Efficiently provide the location information in a overlay network
- Add mechanisms to efficient locate radio devices based on location
- Create a lightweight solution to fit all types of radio devices

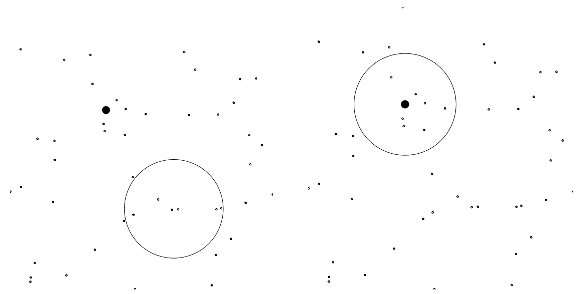
The rest of the paper is organized as follows. Section II discusses work related to the solution presented in this paper. Section III explains our solution - ObeSA, and how we provide and exploit the location information. As a part of ObeSA, we introduce our own geocast mechanism, distance flooding and compare it to existing well-known geocast flooding mechanisms. In Section IV, we explain our implementation, the simulation settings, and performance metrics. At the end of the section, we look at the results. Finally, Section V concludes our work and provides suggestions for the direction of future research.

II. Related Work

Considerable work has been done to provide the physical location of a node into an overlay network. Wang and Ji have identified the need of a better way to give a node an identification based on its location [5]. In this paper, entering nodes are given a node-id which is logically selected in-between two other nodes based on their locations. The idea of setting the node-id based on the location, and not randomly, or hashed, by the overlay network, is an interesting approach. Unfortunately, their solution only gives us the relative distance between the nodes. In order to calculate e.g. possible interferences prior to a spectrum allocation, we need to know about the absolute location of the physical neighbours. Without this information, we do not know if they are in the area where we (or they) can create harmful interference to each other. For our purpose, this solution will not give us enough information about the location of the node. Compared to other overlay network implementations already presented [6], this paper does not give us added information.

Chiang et al. show the use of the *Location Estimation Technique* (LET) in [7]. They calculate a hash based on prefix of a nodes' location information. The first node which has the LET hash generated identification will represent the area and is defined as a super-peer. Other nodes with same generated identification will connect to their super-peer as normal-peers. The approach of taking the node's location and use it as a lookup key to find other nodes in a specific area is close to what we are seeking. Unfortunately, by hashing the result, there is no easy way to extract this information. The result is a hierarchy-based solution containing two levels. In a large network, we see a need to build additional levels of super-peers to be able to scale with the number of nodes and/or the size of area a super-seed should represent. The most interesting part of this paper, for us, is the idea of injecting the location information into the node-id.

Ahulló et al. have proposed a distributed hash table (DHT)-generic way to make geographical range queries in [8] by using a hierarchy and clustering approach. They support data and traffic load balancing while making it possible to search and retrieve data by storing the data key close to its physical position. The authors suggest having cluster-layers based on world/continent/country down to the granularity level that is required. The proposed solution has shown a useful way to support geographically ranged searches in DHT, while maintaining data and traffic loads. To our knowledge, this work is the closest to what we are solving in our paper: combining overlay network and a mechanism to locate nodes that are of interest in a specific area. However, the authors have yet to address how to implement the location information into the overlay network in an autonomous way. In their example, the nodes have 4 bits of addressing, 2 for prefix and 2 for suffix.



(a) Away from the node (b) Around the node

Fig. 1. Search scenarios

It is unclear how they map a node's real location with this type of addressing.

For a solution to our second goal, we have chosen to look at geocast [9]. Geocast is a mechanism to deliver a message to all nodes in a specified area. Geocast can be divided into two main categories: flooding and non-flooding protocols. Flooding generally generates more messages in the network. By introducing smarter and more controlled flooding techniques, we can exploit the provided location information in the overlay network, get better results than non-flooding and avoid the negative network performance hit introduced by full flooding. Although the non-flooding protocols show great scalability, they require additional network infrastructure. This is often achieved by creating and maintaining a hierarchy. In a fixed network with few nodes entering and exiting, the cost of maintenance is low. However, in a network based on radio devices, even if we do not take mobility into account, radio devices will come and go, thus resulting in a maintenance cost we would like to avoid.

III. ObeSA - Overlay-based enhanced Spectrum Allocation

This work is based on using radio devices that are capable of dynamically allocating radio spectrum. All radio devices have a control channel available for data communication. This channel is used for connecting to the overlay network. Additionally, radio devices must also have knowledge of their location. Location information may be gathered from a GPS (Global Positioning System) device, a network location identifying system or from other methods. For scenarios where the radio devices are indoor, solutions based on RFID may be used [10]. In this section, we first present the scenarios, then the algorithm for creating the node-id based on the location information. Last, we introduce our own version of a controlled flooding mechanism, distance flooding, which exploits the location information provided in the node-ids. It is important to keep in mind that we have focused on a solution that will work on all kinds of radio devices. This is achieved by creating a solution which has low impact on the devices involved, this is to meet the third goal of supporting all kind of radio devices stated in Section I. This implies requirements of less computational power and lower bandwidth in control channels than in alternative solutions.

A. Scenario

To be able to carry out cooperated spectrum allocation successfully, one must coordinate with nodes in the surroundings. Depending on the available frequency and transmitting power, it is easy to compute your own radio working area. All nodes in this area should be identified and coordinated for best results. As shown in Fig. 1, we have identified two scenarios: finding nodes in a area away from the node's location (a) and finding nodes around itself (b). Scenario (a) may be a scenario where the a device is claiming a frequency band in a given geographic location away from itself. Even though this may not be a scenario that first comes to mind, there is still a need to support it. Radio devices that are pre-allocating frequencies in another location will need the support from finding other radio devices away from themselves. Such types of users are described in [4]. Scenario (b) is a more typical scenario for a DSA network. Once the radio device is in need of allocating a frequency, the radio device connects to the overlay network to find all its geographical neighbours. Upon successfully identifying its neighbours, the radio device can coordinate and cooperate to make a better decision about its own spectrum allocation.

B. Providing the location information

As described in Section II, considerable work has been done in terms of providing the location information in an overlay-based network. We have focused on a solution which provides this essential information as efficiently and early as possible. We have chosen to implement the node's location information into the node-id as is done in [7]. However, unlike Chiang et al., we chose to store the location information directly into the node-id without hashing the information. As hashing is a one-way function, there is no way to extract the hashed information. By re-using the node-id as information carrier, no additional overhead is introduced to the distribution of the location information of the nodes. Storing the information in the node-id is also done to keep the overlay implementation as generic as possible. Node-ids from other nodes in the overlay network are available in most of the overlay implementations. With this solution, we are not tied to any specific overlay implementations which in turn may make it possible to use different, compatible, overlay implementations within same network. By storing the location information almost unaltered, keeps the processing requirement when creating and extracting the node-id at a low level.

In our algorithm, Alg. 1, each latitude or longitude results in a string containing 9 characters, combined 18 characters. If we assume that we have 24 characters available for the node-id, we have 6 characters left to fill with other information. As node-ids have to be unique for each node in the network, two nodes at the exact same position cannot have the same node-id. This can be solved by various means such as adding random characters to fill up the remaining free space.

In this paper, we assume that the nodes are equipped with a GPS device. GPS devices use the WGS84 datum [11]. For easier calculations, we use GPS decimal formatted coordinates (latitude, longitude) with 5 decimals. GPS coordinates with

Algorithm 1 Creating a node-id based on location information (latitude and longitude).

Incoming coordinate must contain 5 decimals
Result node-id length is 24 characters

for all i = latitude, longitude **do**
  Initiate $result$
  Delete '.' from i
  if i is a negative number **then**
    Set the first character in $result$ = 2
  else
    Set the first character in $result$ = 1
  end if
  Calculate K numbers of 0 to add in the $result$
  $K = (18/2) - i.length$
  Concatenate $result$ with K 0's and i.
end for
Concatenate $result_{latitude}$ and $result_{longitude}$
Fill the empty free space with random characters

5 decimals have an accuracy of approximately 1 metre. A GPS decimal formatted coordinate varies in two areas: A) The latitude and/or longitude can be a negative decimal. B) The integer part of each decimal can contain 1 to 3 integers. As shown in Alg. 1, variance A in handled by using the first character to identify whether the decimal is negative or not. A different number of integers, variance B, in a decimal is handled by filling the "missing" integers with 0's.

In order to explain how we calculate the node-id based on the location information using Alg. 1, let us take a node located at Kjeller Airport as an example. Kjeller Airport's GPS decimal formatted coordinate is 59.96944, 11.03888. Since the latitude is not a negative number (59.96944) and the integer part contains only 2 characters (59), we concatenate the latitude with 10. The first character identifies that the latitude is not a negative number. The second character is 0 filling the "missing" integer. The result is 105996944. Likewise, for the longitude, we get 101103888. The result is: 105996944101103888 plus 6 random characters.

By concatenating the latitude and longitude, and adding different paddings, there is very little computation required in this algorithm. As with the generation of node-id, the extraction of location to any given node-id is straightforward. We would like to stress that we have deliberately created a solution with as low requirements as possible, which resulted in this algorithm to store and extract location information in the node-id.

C. Geocast

Flooding, as a general opinion, scales poorly in large networks. Because radio devices have a certain transmission capability, frequencies have different capabilities, the radio coverage of a device is known and limited to the surrounding of its operation area. Based on this fact, and the location information now provided in the node-id, we believe that using a controlled geocast flooding protocol is suitable for our solution.

To study and find a geocast mechanism for our ObeSA

 ISBN: 978-1-61208-123-6

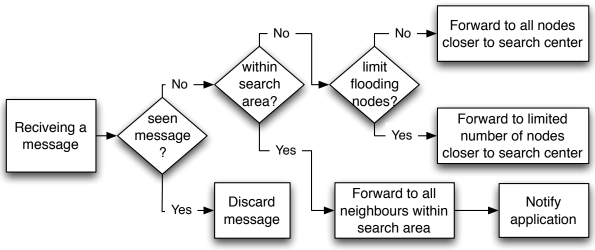


Fig. 2. Distance flooding protocol. Floods to all nodes closer to search centre, floods within search centre.

solution, we have identified 3 different types of geocast flooding protocols:

- ***Simple flooding*** - This is the simplest implementation of flooding. Each node sends the message to all known neighbours. If the message is already seen, the node will discard the message. Each node has to check whether it is within search area or not. This type of flooding does not require or is able to exploit location information if available.
- ***Unicast Routing with Area Delivery (URAD)***- This protocol has been named and simulated in [9]. Based on location information available, URAD uses regular unicast towards its search area. Once the message has entered the search area, the message will be flooded to all nodes in the area. We implemented URAD to unicast (forward) the search message to the node closest to the search *centre* before flooding in search area. This is done to create a direct path as possible towards the search area.
- ***Distance flooding*** - This is our own optimization (Fig. 2. As URAD uses unicast towards an area, we argue that forwarding to multiple nodes closer to the search centre than itself yields a better result. Further optimization of our proposal is to limit number of nodes a message should be forwarded to. We will present the results from these simulations later in this paper, but a deep analysis from these simulations are outside the scope of this paper.

IV. Simulation and results

This section explains how we implemented and which settings we used for our simulation. Then we will look at the performance metrics of the simulations. Finally, we look at and discuss the results.

A. Implementation and settings

We have used the PlanetSim simulator [12] to implement a small geocast mechanism on top of Chord[13]. We are using Chord as the overlay network, because we wanted a simple and well proven overlay network as a base for our solution. As stated before, there should be no problem to use any other overlay network implementations instead of Chord.

The simulated scenarios are flat earth models with an area of 1000 x 1000 metres, with one metre as lowest resolution. All scenarios are pre-generated and re-used for each geocast protocol. For each scenario, the node initiating the search and the search areas are set. Simulations have been run for both scenarios: the search area around the search node and search area away from the searching node. The search area is implemented as a circle, defined by position (latitude, longitude) and a search radius. The search radius is set to 100 metres. The network consists of 100 to 1000 nodes. All nodes are static and are connected to the same overlay network. The displayed results are the average of 100 simulation runs for each protocol in each scenario.

We used an Intel Core 2 Quad Q9400 CPU equipped with 3 GB RAM. The operating system is Fedora Core 12. The simulations takes approx. 7.5 hours to finish.

B. Performance metrics

The performance metrics are based on three parameters: percentage of identified nodes, network load, and number of hops before reaching the last node in a search area:

- ***Percentage of identified nodes*** - is used to measure how good the protocol is to identify all the nodes in the search area. For geocast, this is a very important metric due to the lack of delivery acknowledgement. We need to get as accurate list of nodes as possible of interest in the search area. The higher percentage, the better.
- ***Network load*** - is used to measure how many messages are sent in the overlay network. The lower the number, the better.
- ***Number of hops before reaching last node in the search area*** - is used to measure how fast a search can be finished. Given that all nodes have the same quality in terms of control channel, a lower number means less time (hop) it takes to reach all the nodes within a certain search area.

C. Results

The results shown in Fig. 3, Fig. 4 and Fig. 5 of the distance flooding protocol are with no limit. This means that a node will forward to *all* nodes it knows about that are closer to the search centre than itself. In these figures, the dotted graph lines represent the result of simulations in scenario a (search away from the node). The results from both scenarios in all three algorithms are quite similar, however, the scenario when the search area is around the node (scenario b) has a weaker performance. The reason for this is how Chord selects which neighbours to keep in its finger (routing) table by storing neighbours spread over the whole network.

Fig. 3 shows the simulation results of the percentage of identified nodes in a search area. We see that for both scenarios, simple flooding is performing as expected. It identifies all the nodes, which is not surprising for flooding in a stable network where all nodes are known. Studying the URAD protocol, we see that the more nodes there are in the network, the weaker it performs. This result was expected, and is similar to what has been presented in [9]. Our own protocol, the distance flooding performs very well. When the search area is away from the node, distance flooding has similar results to simple flooding.

 ISBN: 978-1-61208-123-6

In the other scenario, it has a weaker performance. Even though it does not have a perfect performance, it performs much better than the URAD protocol.

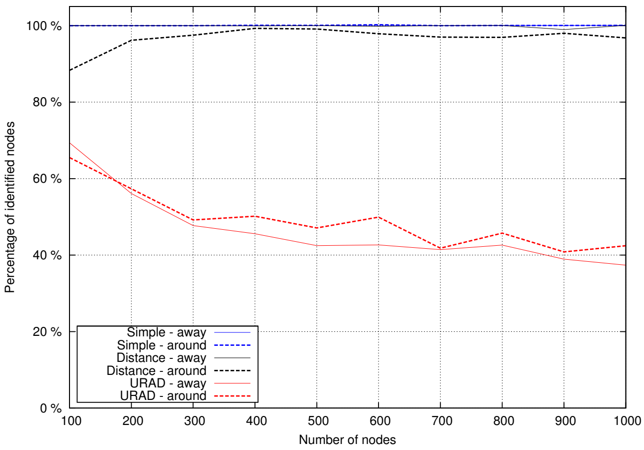


Fig. 3. Percentage of successfully identified nodes in search area.

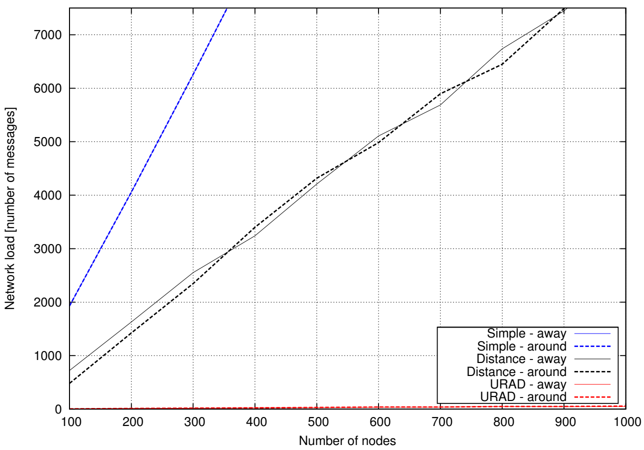


Fig. 4. Total number of messages sent in the network when flooding a message.

The results in Fig. 4 are quite clear. Simple flooding has the highest load on the network. Distance flooding has approx. 60% less network load than simple flooding and the URAD protocol has the best performance. Because URAD uses unicast towards a search area before flooding in the area, there is very little overhead when routing the message towards the search area. Combined with a routing table containing neighbours over the whole network, which is available in Chord, it does not take many hops before the message arrives at a node in the search area away from the searching node.

Fig. 5 shows the results of how many times a message has been forwarded before arriving at the last node (which is in a search area). Again, simple flooding is the worse performer and the URAD protocol is the best. The distance flooding is in the middle, approx 25-30% better than simple flooding.

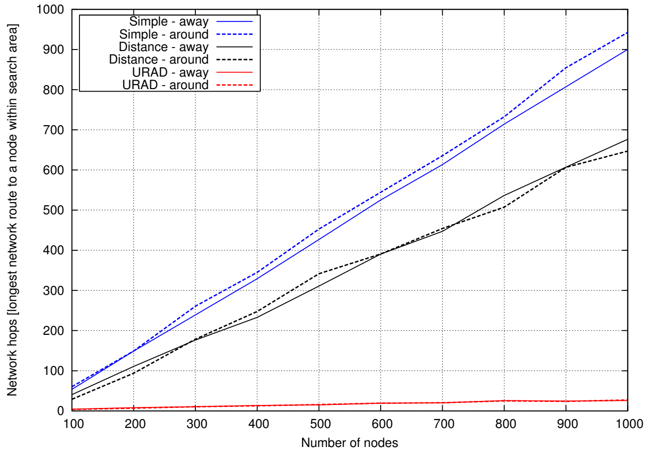


Fig. 5. The longest path a message takes to arriving the last node in search area.

The non-linear and similar results in how simple and distance flooding perform, are the result of the already mentioned Chord's way of selecting nodes in its finger table.

Except for the results in Fig. 3, the result graphs are quite linear. The more nodes there are in the network, the more messages and longer time it takes to identify all the nodes. Even though the URAD flooding protocol is performing extremely well in terms of network load and number of hops before arriving at the last node, the percentage of successfully identifying all nodes within a search area is too low.

Even though our distance protocol has a higher load on the network and takes longer time (as in number of hops) before arriving at the last node than the URAD protocol, the performance of identifying all the nodes in a search area outperforms the URAD protocol. Based on the results, we advocate that the distance protocol is the best mechanism for our solution, ObeSA.

As mentioned earlier, the distance flooding protocol can be further optimized by limiting number of nodes a message should be forwarded towards a search area. The original idea of the distance flooding is to forward to *all* nodes that are closer to the search centre than itself. The term *limit* used in the distance protocol context means that a limit number is set to control how many nodes the distance protocol should forward to. E.g. if the limit is 2, each node should only forward the message to maximum 2 nodes that are closer to the search center than itself. We have simulated the protocol using the same scenarios used earlier, with different number of limits ranging from 2 to 5. If the limit is set to 1, distance flooding will perform exactly like the URAD protocol, which is unicast towards the search area before flooding.

The results in Fig. 6 and 7 indicate that by limiting the number of neighbours a node should forward a message, the network load can be greatly lowered and the percentage rate of successfully identified nodes is still at a much better level than the URAD protocol. The results show that if the limit is set to 2, we see a great improvement in the percentage of identified nodes in a search area. URADs lowest result is at 38% while the distance protocol has the lowest result at 73%. Looking at the network load, we see that URAD in scenario with 1000 nodes uses 26 messages to identify all nodes in the search area, while the distance protocol with limit = 2 uses 72 message. The number of messages is still quite low compared to simple flooding and no limit distance flooding and still maintaining a good percentage of identified nodes.

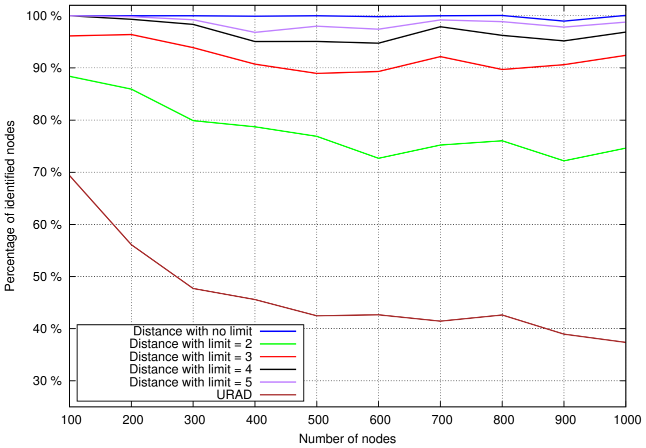


Fig. 6. Percentage of successfully identified nodes in the search area.

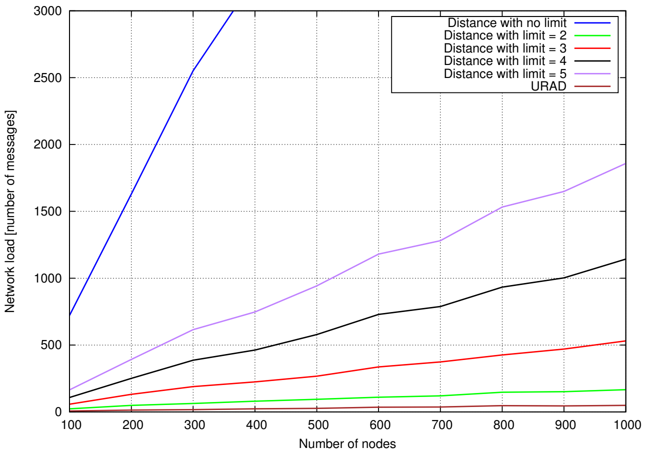


Fig. 7. Total number of messages sent in the network when flooding a message.

V. Conclusion and Future Work

The solution presented in this paper is to our knowledge one of the first examples of combining an overlay-based network and geocast providing a system for cooperation of dynamic spectrum allocation. Our main contributions are: A) creating an algorithm that takes location information and creates a valid node-id. By using the node-id as the information carrier, we have introduced an efficient way of exchanging this information. B) showing how the provided information can be exploited. Here we have created a new geocast flooding protocol called distance flooding. The result of the simulations show that by exploiting the provided location information, network performance can be greatly reduced while still keeping the accuracy of identifying nodes in the search area. Combining the distance flooding protocol and the location information stored in the node-id in the overlay network, ObeSA goes beyond the state of the art in DSA in terms of providing a cooperation platform for spectrum allocation.

As mentioned in Section IV-A, we have used a flat earth model in our simulations. In other scenarios, e.g. urban areas with high buildings, we will see situations where multiple nodes are located at the same position, but at different elevations. To distinguish between these nodes, we can use some of the 6 available characters randomly filled when creating the node-id in Alg. 1. Other information that might be interesting to store in the node-id could be the operation frequencies of the radio device.

The main limitation of this study is the lack of mobility. Even though it is not a requirement that radio devices must be mobile, radio towers or TVs, mobility support should be studied and added to ObeSA. Further study and optimization of the distance flooding protocol is also considered.

References

[1] J. M. III, “Cognitive radio an integrated agent architecture for software defined radio,” *PhD dissertation, Royal Institute of Technology (KTH), Sweden*, pp. 1–313, May 2000.

[2] FCC, “Report fcc-10-174,” pp. 1–88, Sep 2010.

[3] T. Ulversoy, T. Maseng, T. Hoang, and K. J., “A comparison of centralized, peer-to-peer and autonomous dynamic spectrum access in a tactical scenario,” *Military Communications Conference, 2009*, pp. 1–8, 2009.

[4] T. Hoang, M. Skjegstad, T. Maseng, and T. Ulversøy, “Frp: The frequency resource protocol,” *12th IEEE International Conference on Communication Systems 2010*, pp. 1–5, 2010.

[5] S. Wang and H. Ji;, “A topology-aware peer-to-peer protocol applicable to wireless network,” *Network Infrastructure and Digital Content, 2009. IC-NIDC 2009.*, pp. 1004 – 1009, 2009.

[6] M. Castro, P. Druschel, Y. Hu, and A. Rowstron, “Exploiting network proximity in distributed hash tables,” *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, pp. 52–55, 2002.

[7] H.-K. Chiang, H.-W. Chen, and F.-L. Kuo, “Locality support for mobile p2p network,” *IWCMC ’07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*, Aug 2007.

[8] J. Ahullo, P. Lopez, M. Artigas, and A. G. Skarmeta, “Supporting geographical queries onto dhts,” *Local Computer Networks, 2008. LCN 2008.*, pp. 435–442, 2008.

[9] C. Maihofer, “A survey of geocast routing protocols,” *Communications Surveys & Tutorials, IEEE*, vol. 6, no. 2, pp. 32 – 42, 2004.

[10] S. Saab and S. Nakad, “A standalone rfid indoor positioning system using passive tags,” *Industrial Electronics, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2010.

[11] N. Imagery and M. Agency, “Department of defense world geodetic system 1984, its definition and relationships with local geodetic systems,” *Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems*, vol. NIMA Technical Report TR8350.2, pp. 1–175, Jan 2000.

[12] P. Garcia, C. Pairot, R. Mondéjar, and J. Pujol, “Planetsim: A new overlay network simulation framework,” *in Software Engineering and Middleware*, Jan 2005.

[13] I. Stoica, R. Morris, D. Karger, and M. Kaashoek, “Chord: A scalable peer-to-peer lookup service for internet applications,” *in Proceedings of ACM SIGCOMM*, Jan 2001.

 ISBN: 978-1-61208-123-6