*IoT Device Discovery using a Social Network Approach*

James Sunthonlap

Department of Computer Science

California State University, Los Angeles

Los Angeles, USA

Phuoc Nguyen – (Shay)

Department of Computer Science

California State University, Los Angeles

Los Angeles, USA

***Abstract*—The Internet of Things (IoT) is one of hottest and most intriguing topic in today’s society. More and more devices are connected to the internet and it may surprise you to learn that there are more connected devices than there are connected humans [4]. The pervasiveness of smart devices, connected things, raise a major concern: how does a device “discover” another device? How do we plan to handle the eventual scalability issue? We believe that a “Social Networking” approach towards device discovery will help provide a more scalable solution towards device discovery, helping us to fully take advantage of the vison IoT promised us [1]. In this paper, we will propose a model that represents human social relationship applied to IoT devices. We are doing an extension on past research done towards social networks and applying it towards the IoT domain. We will then test and evaluate our model with a more traditional centralized and decentralized approach.**

# Background

Internet of Things (IoT) consists of many interconnected devices with sensors and actuators all connected through the internet. These devices can sense, control, evaluate, analyze, and make decision autonomously, alone or collaboratively with the aid of other devices. Since IoT has many applications such as smart home, smart city, healthcare monitoring, and logistics, its popularity has surged in recent years.

In the near future, more and more IoT devices will be connected to the Internet, making the number of IoT enabled devices extremely large. As more devices connected to the internet continue to grows, the amount of traffic is expected to increase as well. According to Cisco, internet traffic by non-PC will increase to just under 70% in 2019 [3]. Device discovery, the abilities of devices to properly discover and communicate with each other, will be increasingly more difficult as more things become “smart” and the searching space grows, posing a future scalability problem.

One of the most interesting features of a social network is its ability to connect people together. Users can *traverse* through other users’ *profile* to find an old childhood friend, a new romance, or even a job. Users can *establish relationship* with people who are not in their close proximity. Users can *discover* people with common interests who may not be visible normally. Similarly, to address the possible future problem of device discovery of a massive scale, an approach that mimics the behavior of social networks and applies them towards the Internet of Things would be an idea way to address the scalability and service discovery problem.

# Related Works

Current solutions towards device discovery involves a centralized and decentralized network topology. The centralized network involves a center node being connected to all other nodes. This center node is a server and the other nodes are IoT devices connected to that server. The server will handle requests from all other devices. Since the trend is an ever-increasing amount of IoT devices, this means that the server will have to handle an ever-increasing number of requests and responses. Another problem with the centralized approach is the fact that the center node represents a single point of a failure. Lastly, storage requirements will be a problem since storage index is concentrated on a single node [2].

The decentralized approach involves only nodes being connected to other nodes; there is no single master or center node. However, this means that in order for devices to discover other devices, these devices will need to flood the network with many requests rather than contacting a center node [2]. This reduces network bandwidth efficiency. Since devices would be pinging other devices arbitrarily, this would reduce the success rate of a device finding a device it is looking for. Many pings to devices are involved, which means a longer discovery time.

# Proposal

The belief that any two people, or devices, can be connected together by simply exploring their social network with only their local information (friends, families, etc..) is based on the works of social psychologist Stanley Milgram [5]. In his famous 1967 study, he observes that people are tied by a short chain of connections. A more recent Facebook research confirms this observation when they concluded that the average person among their 1.59 billion active user base is only separated by 3.57 persons [6]. IoT devices that exhibit human relationships (friends, co-worker, family, etc.…) could therefore be used to navigate through their “social network” in the same way that people do.

In this paper, we propose a model for scalable device discovery using concepts from “Social Networks”. In our model IoT devices can establish meaningful relationship with other devices and thus they will be more likely to find their required service by searching amongst their “friends” as oppose to searching through the global network. For example: Suppose “Jim” requires assistance with his printer because it is not working. Jim works with Ann. Ann is married to Bob who can fix printers. Through his relationship with Ann, Jim can establish a relationship with Bob and acquire his service. This social relationship could be modeled and applied to IoT devices as well. The “social” relationship that devices can form reveals a structure that can be navigated. Through this structure, an IoT device can independently crawl and browse their “social network” and search for or broadcast a service. Devices will become more visible to other devices. A device is more likely to find what it’s looking for if it searches amongst its “friends” or “social network” and therefore we expect device discovery to be more scalable and efficient.

To define and model social relationships for intelligent objects we first must consider which relationships are most important to a typical person. Intuitively the three that comes to mind are family, work, and friend relationships.

An intelligent device’s family relationship can be a group of devices whose relationships will not change, such as their manufacturer. These (family) devices may or may not work together. An Apple iPhone may or may not work with an Apple TV. Devices’ working relationship can be described as two devices that *definitely* work together to provide a service. The relationship could change frequently. An example could be a user’s personal heart rate sensor that provides data to another application which in turn is used by a doctor to provide the appropriate care. An IoT device’s friend relationship can be described as objects owned by the same person or company as they are most likely to interact with one another. They tend to share a common theme. A smart rug and light bulb are wildly different in functionality but they can work together in a smart home to turn on the lights when the owner comes home. Like the co-worker relationship, these relationships could change frequently. Devices’ relationships are established and updated based on their activities and features.

Devices can form relationships with each other, but not every relationship is the same. A relationship can be bidirectional (the Facebook *friendship* model) where both objects are required to subscribe or approve each other. Alternatively, a relationship could be unidirectional (the Twitter *following* model) if only one device subscribes to another device service or data while the other one does not.

Each device is autonomous in the sense that they can establish social relationships autonomously. The social aspect of IoT devices gives it the ability to form networks with other devices. Together, each device can become socially aware and benefit mutually from networking. This will enable smart devices to learn about other devices and then act on this information. Devices will use the “social network” the same way people would. People can use it to become more popular, devices can use it to publish their data or services. People can use it to find old friends, devices can use it to discover data or services. People can use it to find new friends, devices can use it to discover new data or services. Like humans, IoT devices’ social network changes frequently. The social aspect allows machines to dynamically create social connections with each other and work together to solve problems.

After we develop and implement a model we plan to test the IoT social network. Our evaluation will be based on the following criteria: discovery time (how long does it take to find a service), network traffic (the number of message issued until a service is found, success rate (percentage of success). Furthermore, we plan on comparing these results to a more traditional approach in IoT, a centralized and decentralized approach. We want to be able to identify each approaches’ strengths and weaknesses and most importantly, identify if the Social Network of Internet of Things is feasible.

A lack of a common communication protocol and the heterogeneity of devices are some of the biggest challenges to overcome. Most IoT devices are created as a standalone application. Thus, it may be difficult to build a social network amongst these devices as these applications can vary wildly. Human social networks differ greatly from devices in this aspect. All humans are the same while devices are not. Social network navigability may be a concern as the network evolves and each device develop more relationships. How can we ensure that devices select/prefer the most reasonable links? This could lead to higher computation complexity. How much should we seek to represent? Too much may make implementation extremely difficult, but too little may not yield accurate information. The model uses device profiles to discover other devices, which leads to another issue: *how* do we emulate service discovery in our simulation. What existing algorithm could we use? A discovery algorithm will heavily influence our evaluation and implementing one will be vital to our research. Another potential issue we face are isolated networks. Like people, a device or a network of devices may be so remote that they have no contact with the outside world, possibly yielding limited benefits from social networking.

# Methodology

Rather than use an off-the-shelf simulator, we opted to develop our custom simulator software application. The application allows high flexibility in configuring the network environment, test methodologies, and search algorithm. After all, these three factors are interdependent. For example, a network environment with only a small number of nodes might yield insignificant performance differentials across the centralized, decentralized, and social networks as the search algorithm in a decentralized network need not traverse very far.

## Generating the Network Environment

The method in which the social network is generated has a significant impact on its performance. We generate randomized and scale-free networks for centralized, decentralized, and social network topologies. The randomized network is random in that there is no order of preference when a node gets added to another node. A social network in the real world is often similar to a scale-free network. A scale-free network is one in which there are a few nodes with a high number of connections, with the rest having very few connections [8]. This is because a scale-free network follows a power law, where the fraction of nodes having k connections as given by (1),

(1)

, where is roughly between 2 and 3 [7]. The number of extremely popular nodes is small because as nodes get added to the network, they will have a higher probability of getting added to existing nodes with more connections. As time goes on, the popular nodes get more and more popular.

A script is run in order to generate nodes placed into a topology that mimics a scale-free network. Each of the nodes in these networks contains two random features. For example, two such features might be smart lightning and smart parking. The script will also generate neighbors for each node which follows the power law. Each of these nodes will be placed into centralized, decentralized, and social networks. The only difference between the nodes across the three networks will be the nature of relationships with each other. For example, in the social network, a node might have three neighbors, with a family, friend, and co-worker relationship link respectively. The same node with have the same neighbors in the decentralized network but there will exist no relationship links between that node’s neighbors. In the centralized network, that node will only have one neighbor, which is the master node. Each of the other two neighbors might be the master node or a slave node.

## Performance Evaluation

Every network will undergo a performance evaluation that tests the average latency, success rate, and average number of hops across one thousand searches. Each of the searches will start from a random node and attempt to find a random, but existing feature. The search over each network evaluation will contain the same source node and feature searched in order to maintain fairness. Each of the networks will be tested with 20k and 50k nodes. For both network sizes, there will be multiple tests performed such that the number of features are ten percent of the total number of nodes. There will be five such feature amounts in increasing multiples of five. For example, in the twenty-thousand node network, we will test centralized, decentralized, and social networks with 2k, 4k, 6k, 8k, and 10k features. This will enable us to see how the number of features affects the performance, success rate, and average number of hops for each network. Testing across multiple feature configuration also allows us to determine the optimal number of features a social network should have.

## Search Algorithm

The social network’s search algorithm performs a breadth-first-search (BFS) whereby each node along the search path will ping their neighbors for the feature in preferential order of a neighbor’s centrality, diversity, and clustering co-efficiency. The centrality of a node is the number of neighbors it has. Diversity is the number of unique relationships with its neighbors. Lastly, the clustering co-efficiency of a node is a measure of how close-knit its neighbors are. More formally, the clustering co-efficient of a node is given in (2).

(2)

, where is an edge from vertex *j* to vertex *k,*  is the set nodes or vertices in the network, *E* is the set of edges in the network, and is the node *i*’s centrality. The clustering coefficient is involved in the BFS algorithm since social networks often have groups of nodes that are tightly clustered together. The algorithm will thus prefer to ping a neighbor whose neighbors share higher amounts of links between them.

The decentralized network’s search algorithm consists of a breadth-first-search without preferential order. The neighbors of each node are searched in no particular order during a BFS of the decentralized network because they are randomly ordered. Lastly, the search for a feature in the centralized network involves a non-master node contacting the master node, which in turn will contact another node if necessary. A search in the centralized network will thus involve at most 2-3 nodes depending on if the source node of a search is the master node or otherwise.

# Results

## Nodes Contacted

One of our main interests in this research is to measure, during a series of searches, a randomized networks’ average number of devices contacted before the device with the desired feature is discovered, and make comparisons between the social network and traditional decentralized approach. The fewer the number of required nodes to discover the node with the target feature, the better. A smaller value means less computational time and smaller number of requests flooding the network. According to Fig. 1 and Fig. 2, it appears that both the number of devices and its feature diversity, the number of unique features of the devices, play a very significant role in these results. The more devices there are and the more diverse those devices are, the better the social approach will perform relative to a decentralized approach in terms of average nodes contacted. Our evaluation considers only successful searches. The results are promising because it shows that at a very basic level, a social network of devices can effectively leverage social relationships to discover other devices in a way that decreases number of nodes contacted.

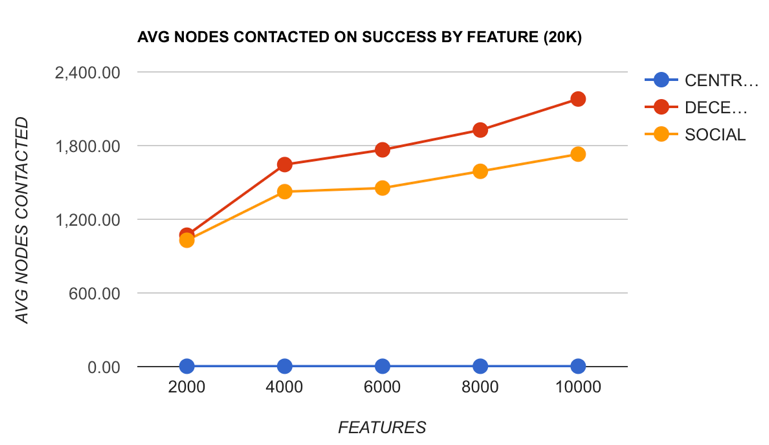


Fig. 1. Average number of nodes contacted in the randomized network containing 20k nodes.

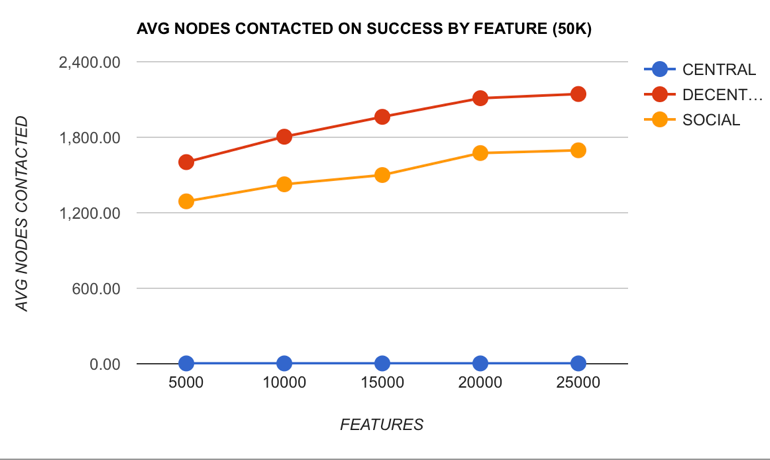


Fig. 2. Average number of nodes contacted in the randomized network containing 50k nodes.

## Success Rate in Random Network

While the outcome from the average nodes contacted was exciting, this was short-lived when looking at the success rate metric. The alternative approaches to a traditional centralized method are exciting and promising but they are not reliable. As seen in Fig. 3 and Fig. 4, as the number of unique features and or devices increased in the network, the success rate of discovering a device decreased dramatically in both the social and decentralized approach. Note that the y-axis of the graphs is not on the same scale. A discovery search is considered to be successful if a device discovers another device with the desired feature before *Time to Live,* the fixed time in millisecond, expires. Our raw number didn’t worry as much because our standard is probably too high and it could be adjusted. What is perhaps disappointing is the trend: as the network grows and the devices become more diverse, failures are more likely to happen. This is disappointing because we are trying to address a near future problem of having billions of devices in a network, but this issue could be heavily attributed to how we manually structure our network.

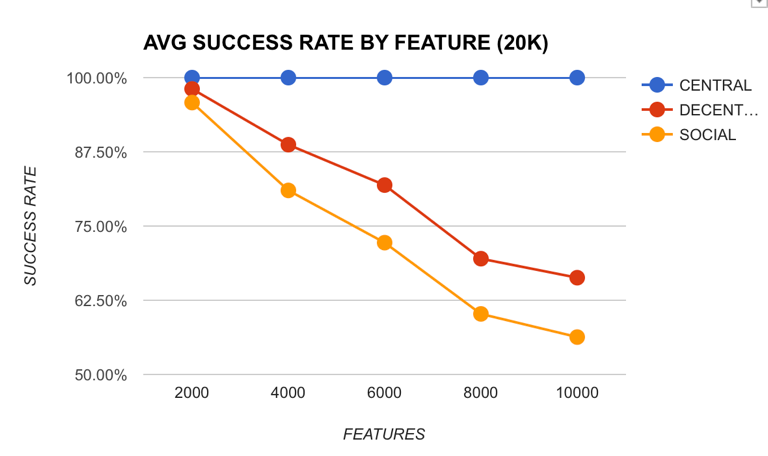


Fig. 3. Average success rate in the randomized network containing 20k nodes.

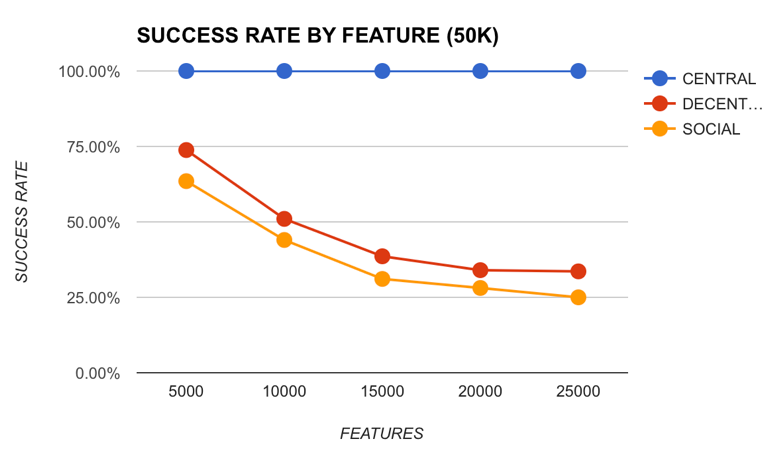


Fig. 4. Average success rate in the randomized network containing 50k nodes.

## Success Rate in Scale-Free Network

Although the success rate drop observed by running the simulation in a scale-free network is not as dramatic as that of the random network, the trend is still similar and the unsuccessful rate is still significant; more devices and features leads to lower success rates. This is evident in Fig 5. and Fig 6. We believe that in both cases the result is due to how the network is structured. Since we manually created the network, it may not represent an ideal social network based on relationships and features, hence the disappointing results.

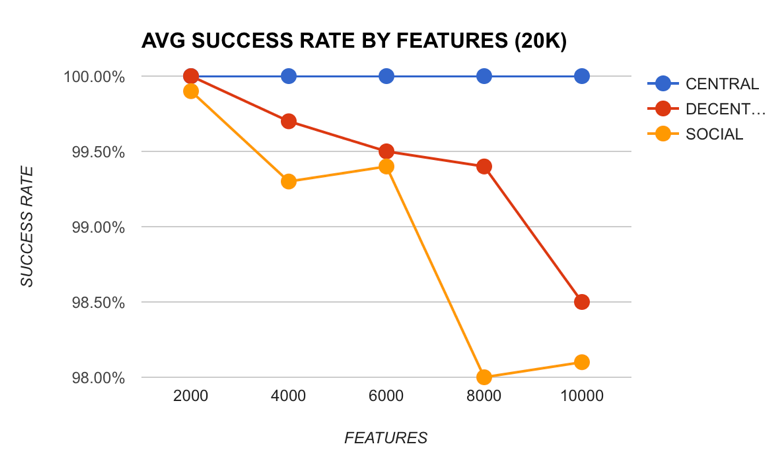


Fig. 5. Average success rate in the scale-free network containing 20k nodes.

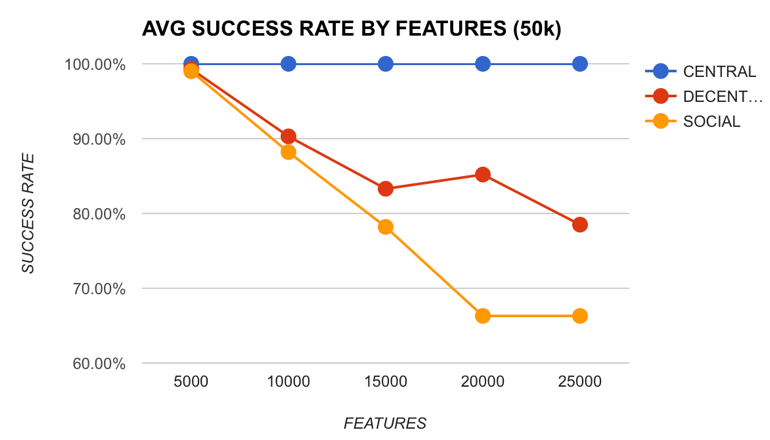


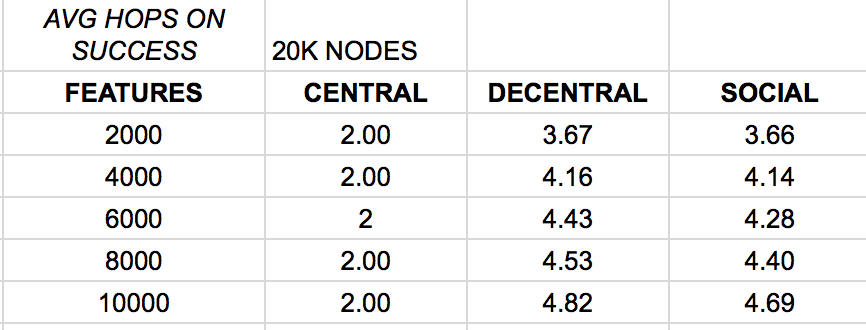
Fig. 6. Average success rate in the scale-free network containing 50k nodes.

## Average Device “Hops” and Latency in Random Network

The average numbers of hops (or edges) from a device to another discovered device in a random network, and similarly the average latency between them, yields a very interesting result in terms of social network theory; all devices in the network are separated by, on average, four other devices regardless of the number of devices in the network. Table I shows this result for 20k nodes. Table II shows that the same result persists even with 50k nodes. For example, if device A wanted to reach device E, device A have to contact, on average, 4 other devices (A to B, B to C, C to D, and D to E). As the devices becomes more diverse, the number of hops increases slightly.

##### TABLE I

##### Average number of hops in the randomized network with 20k nodes



##### TABLE II

##### Average number of hops in the randomized network with 50k nodes

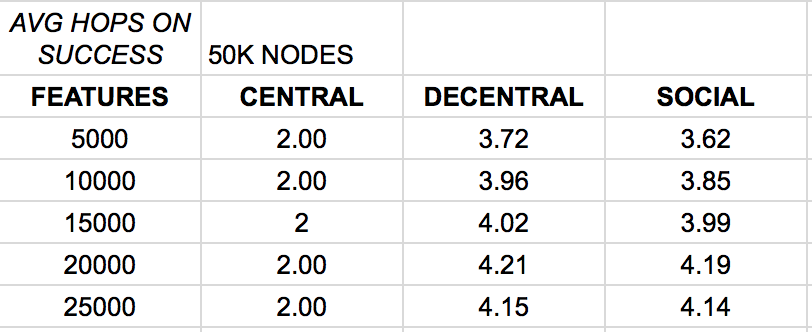
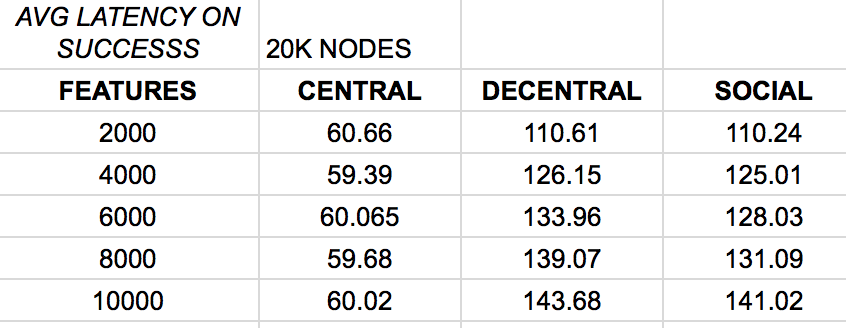


Table III and Table IV shows the average latency in the randomized networks. The average success latency in milliseconds of a search is quite interesting because the network with 50K devices has a lower average latency than the network with 20K devices. Yet as the number of features increases, the average latency seems to increase for the social and decentralized networks. The reasoning behind this is unclear.

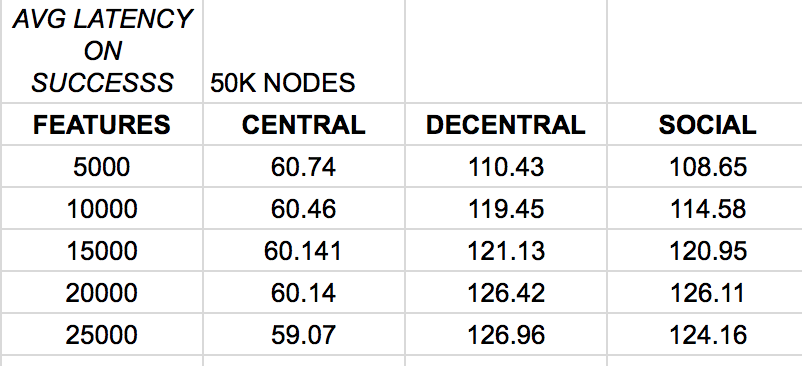
##### TABLE III

##### Average latency in the randomized network with 20k nodes



##### TABLE IV

##### Average latency in the randomized network with 50k nodes



## Average Device “Hops” in Scale-Free Network

As seen in Table V and Table VI, the average numbers of hops in the scale-free network increased from four to six nodes, but the general idea remains the same. Some device A is only separated a by small number devices to some another device B. The difference between the two networks is probably the result of how the group or clusters of nodes are formed. In the random network, each group size is from a fixed range and each device has an equal chance of being part of any group. However, in the scale-freenetwork, each device tends to gravitate towards and connect to devices with a high number of relationships. The end result is there are a few so-called “hub” devices and a lot of small group devices. The key in discovering a device in a scale-free network is therefore to reach the “hub” device. Hence, going from the random to scale-free network exhibits an average increase of two more hops.

##### TABLE V

##### Average number of hops in the scale-free network with 20k nodes

##### TABLE VI

##### Average number of hops in the scale-free network with 50k nodes

Fig. 7 displays a count of the number of devices that make X hops to reach to their discovered device in a scale-free network. It is interesting because it confirms passed research done on human social network in that any one person, or device, in the world is only separated by a few other persons, or devices. This is commonly referred to as six degrees of separation. If a device isn’t discovered in a few hops, then the probability that a search is unsuccessful increases dramatically. Successful searches tend to exhibit similar number of hops and latency. We concluded that unsuccessful searches are result of devices that are relatively isolated from other devices. Thus, they were more likely to violate the time to live limit. Successful searches tend to involve devices that are more clustered together.

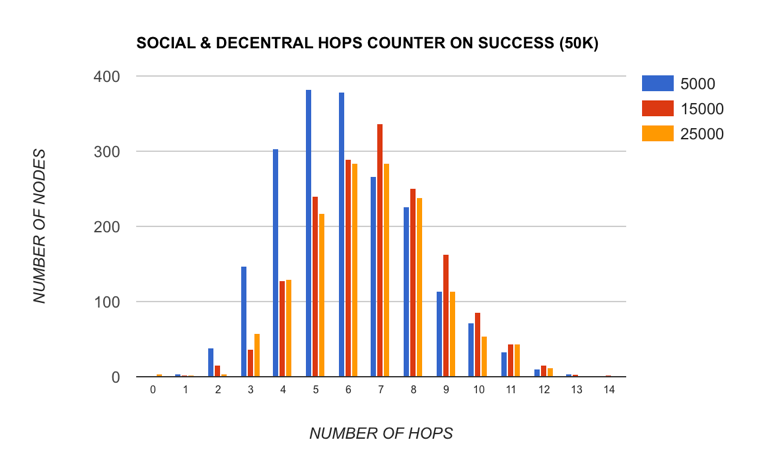


Fig. 7. Number of nodes by number of hops in the scale-free network.

# Challenges

One of the biggest challenges in our research is formulating a so-called “social” network. We tried using an existing social network dataset but we were unsuccessful. We could not find any dataset that contains all the properties that we needed: nodes (IoT device), edges (relationship between two device), communities (types of relationships), and most importantly node properties (IoT device features). As a result, we had to create our own network which was a lot more challenging and complex than initially thought. In fact, it was the most challenging and time consuming aspect of the research. The network generated is not perfect but we feel that our simulation results are still meaningful.

Configuring the values of all parameters that have an impact on the result was another big challenge. For example, how much time should we give to each device to search before we consider it a failure? Each device can cache their result, but how many cached results should they be allowed to keep at most? How many devices should we let a discovery search find? After all, there can be more than one device in the network with the desired feature. How many features should a single device have? A search will naturally perform better if any of the devices in the network can have many features. There are many other parameters involved in creating the networks. The configuration values of these parameters will most likely need to be optimized through further research.

# Conclusion

The goal of this paper is to address the impending scalability problem of IoT device discovery in the near future.

We propose a simple, yet novel approach using the work done in social network research and apply towards the realm of Internet of Things. Devices can establish unique relationships with other devices and through devices’ relationships with other devices, and the relationships of the relationships of other devices, we can form a so-called “social” network and use this network to handle our query and search for devices based on a desired feature.

The results, while not perfect, are still very promising. The social network approach clearly exhibits some performance increases when compare to a traditional decentralized approach where devices merely flood the network with requests until the desired device is found. One of the most exciting conclusions we found is that each device in a network is separated only by a small number of devices (on average of four or six depending how the network is formed). We believe that if a more advanced discovery algorithm is performed on an improved formulated network, the result will be more insightful. To further develop a Social Network of Devices, further future research is required in studying and structuring devices in the network topology and in improving the search algorithm.

##### References

1. P. Mendes, “Social-driven internet of connected objects,” in *Proc. of the Interconn. Smart Objects with the Internet Workshop*, 2011.
2. P. C. Ccori, L. C. C. De Biase, M. K. Zuffo and F. S. C. da Silva, "Device discovery strategies for the IoT," *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, Sao Paulo, 2016, pp. 97-98.
3. “Cisco Visual Networking Index: Forecast and Methodology, 2014-2019,” Cisco, 2015. Available: http://www.cisco.com/c/en/us/solutions/collate ral/service-provider/ip-ngn-ip-next-generation-network/white\_paper\_c11-481360.pdf
4. D. Evans, “How the Internet of things Will Change Everything Including Ourselves,” Web blog post. *Internet of Everything*. Cisco Blogs. May 2011–Sep. 2015.
5. Travers, J., & Milgram, S. (1969). An Experimental Study of the Small World Problem. Sociometry, 32(4), 425-443. doi:10.2307/2786545
6. S. Edunov, C. Diuk, I. O. Filiz, and S. B. Burke, “Three and a half degrees of separation,” 2016. [Online]. Available: https://research.fb.com/three-and-a-half-degrees-of-separation
7. Yaneer Bar-Yam, “Concepts: Power Law”, New England Complex Systems Institute, 2011 [Online]. Available: <http://www.necsi.edu/guide/concepts/powerlaw.html>
8. G. Caldarelli, “Scale-Free Networks,” London: Oxford Univ Press, 2007.