Rayzit: An Anonymous and Dynamic Crowd Messaging Architecture

Constantinos Costa, Chrysovalantis Anastasiou, Georgios Chatzimilioudis and Demetrios Zeinalipour-Yazti

Department of Computer Science, University of Cyprus
P.O. Box 20537, 1678 Nicosia, Cyprus
{costa.c, canast02, gchatzim, dzeina}@cs.ucy.ac.cy

***Abstract*—The smartphone revolution has introduced a new era of social networks where users communicate over anonymous messaging platforms to exchange opinions, ideas and even carry out commerce. These platforms enable individuals to establish social interactions between strangers based on a common interest or attribute. In this paper we present *Rayzit*[1], a novel anonymous crowd messaging architecture, which utilizes the location of each user to connect them instantly to their k Nearest Neighbors (kNN) as they move in space. Contrary to the very large body of location-based social networks that suffer from bootstrapping issues, our architecture enables a user to always interact with the geographically closest possible users around. We establish this communication using a fast computation of an All kNN query that generates a dynamic global social graph every few seconds. We present motivating application scenarios and the detailed back-end architecture that allows *Rayzit* to scale. We have collected and analyzed data from the interactions of thousands of active users and confirm our claims.**

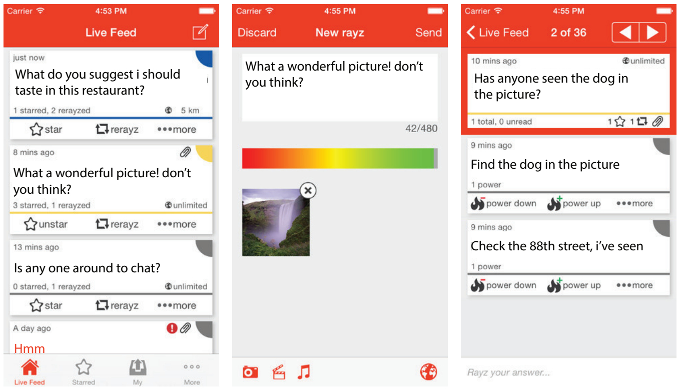
***Keywords*—Crowd, Social, Mobile, Microblogging**

I. Introduction

During the last years, social networks have transformed the way individuals express and publish their opinions, feelings or thoughts on various topics [19]. A new trend even shows that social networks are starting to become the main commerce platforms, instead of web-apps and online shopping carts[2]. Combining the above with the increased amount of sensors on portable devices [16], gives the opportunity to extend the functionality and the intelligence of social networks. This is already illustrated by real world examples like *EARS*, a real time decision support system for earthquakes based on social networks [2].

The location of a user very often defines their interests and inquiries, therefore, popular location-based social networks with billions of users (i.e., Foursquare.com, YikYakapp.com, goTinder.com) have emerged. Furthermore, anonymous networks have been on the rise, where users do not need to create a profile in order to communicate with others, rather they stay anonymous and have an automatic communication link to a subgroup of users based on some criteria (e.g., location).

In this paper, we present an anonymous dynamic crowd messaging architecture, coined *Rayzit*, which is based on generating a kNN network among users by connecting each user to their k geographically closest peers at each moment. Traditionally, social networks like Facebook.com, Twitter.com and others are created based on explicit friendship links that yield a static social graph. Such a static graph requires time to become interconnected, changes very slowly and its links require user decision and actions to be created and destroyed (i.e., request or accept friendship, un-friend). In contrast, *Rayzit* uses a kNN graph, which is developed automatically using the current locations of users in the system. This dynamic graph guarantees k links for each user instantly without any user decision or action [13], leading to communication network that is as diverse as one's physical neighborhood.



What do you suggest i should taste in this restaurant?

What a wonderful picture! don't you think?

Is any one around to chat?

Hmm

What a wonderful picture! don't you think?

Has anyone seen the dog in the picture?

Find the dog in the picture

Check the 88th street, i've seen

Fig. 1. **Screenshots of the Rayzit app**: (Left) Live feed of rayz messages; (Center) Sending a new rayz along with the power bar and an attachment (image, audio, video); and (Right) a set of replies to a rayz.

Our solution utilizes a combination of state-of-the-art computational techniques and crowdsourcing concepts to provide a new compelling social interaction experience through our award-winning app[3], allowing a user to instantly engage in anonymous conversations around them, and using crowdsourcing mechanism to determine the inappropriateness or popularity of posts. Additionally, we have deployed the *Rayzit* app and collected interaction data over a 1 year period from thousands of users, and present our findings through experimental analysis conducted on this data. Such an instant network provides opportunities for advanced location-based services, e.g., location-based marketplace and marketing [10] or disaster relief [22].

[1] Rayzit: `http://rayzit.com/`

[2] Chris Messina "Conversational Commerce": `http://goo.gl/xJBolh`

[3] Appcampus: `http://www.appcampus.fi/`

A. Motivating Examples

During a disaster, mobile users can be both victims and rescuers involved in receiving but also providing help from/to their neighboring peers. In such a scenario, constructing a kNN graph can enable communication within the crowd, aid in organizing the crowd, allocating peer-to-peer aid and experts, and distributing tools and medicine optimally. In a real world example, it was the citizen's joint efforts to map the 2012 floods in China that materialized faster and more accurately than the government-sanctioned map[4].

In large ad-hoc events, like cultural festivals (e.g., Woodstock, Old Car enthusiast gatherings, etc.), sporting events, conventions and fairs, demonstrations (e.g., Occupy Wall Street 2011, Egypt Protests 2013, Syria Protests 2014, Rossa Montana Protests Bucharest 2014, Hong Kong Protests 2014, Pakistan 2014, etc.), monitoring the kNN graph centrally and providing communication within the crowd can aid organizing authorities to manage the crowd[5], prevent crowd disasters[6,7], enable new viral marketing strategies, entertainment services[8] and crowd-games[9].

To the best of our knowledge, this is the first anonymous crowd messaging architecture that is utilizing the proximity and the dynamic social environment of a kNN network. With the rise of *conversational commerce*, where commerce is done over messaging services, and crowdsourcing platforms, e.g., TaskRabbit.com, new types of marketplaces will be generated. In this landscape our anonymous crowd messaging platform can offer new possibilities.

The rest of the paper is organized as follows: Section II looks at the related work, Section III introduces our *Rayzit* architecture thoroughly covering each of its modules. Section IV presents our prototype application developed for several platforms (Windows Phone and iOS), Section V provides a small evaluation using *Rayzit* data while Section VI looks into future issues and concludes the paper.

II. Related And Background Work

Here, we highlight background and related work in social networks, Big data infrastructures, crowdsourcing techniques and discuss how a kNN network can be efficiently computed.

A. Social Networks

Classical social networks, like Facebook.com, Twitter.com, Foursquare.com, SnapChat.com and WeChat.com, create a static social graph based on the friendships between user. Friendship links need to be requested and consecutively accepted/rejected in order to be formed, causing the social graph to be built very slowly.

In contrast, applications like Secret.ly, YikYakapp.com, and Whisper.sh create dynamic social graphs based on the user's location [18]. YikYak connects you with the users within a 10 mile range and preserves anonymity. Secret and Whisper have no distance limit, but they store information about the user's profile (i.e., user id) decreasing user's anonymity. WeChat also has a feature that allows you to greet other WeChat users that are around you, which compromises the anonymity of somebody aiming to stay anonymous.

TABLE I. Taxonomy of anonymous social net applications

Application	Distance	kNN	Anonymous	Location
Facebook	No	No	No	Yes
Twitter	No	No	No	Yes
Foursquare	Yes	No	No	Yes
SnapChat	No	No	No	Yes
WeChat	Yes	No	No	Yes
Secret	Yes	No	Partial	Yes
YikYak	10 miles	No	Yes	Yes
Whisper	Yes	No	Partial	Yes
Rayzit	Custom	Yes	Yes	Yes

Rayzit is the only application that creates a dynamic social graph based on kNN and a customizable distance cutoff setting, guaranteeing at the same time that you will always be connected no matter how far your neighbors are. In addition, *Rayzit* preserves the anonymity and privacy of the user, since it does not require an account and discards any information regarding the status of the social kNN graph.

In order to comprehend the difference between the aforementioned social networks we present a small taxonomy in Table I. We choose all the important characteristics that apply to current social networks. In addition, we include the kNN feature that only applies to *Rayzit*.

B. Crowdsourcing

Crowdsourcing is a process that involves outsourcing tasks to a distributed group of an undefined crowd through an open call for monetary or ethical benefit. There are many examples of crowdsourcing systems, e.g., MTurk.com, Topcoder.com, CrowdFlower.com, and TaskRabbit.com. One of the best crowdsourcing system examples is the ESP game, where the users implicitly collaborate to label images as a side effect while playing the game [8].

It is known that social networks need a mechanism to detect intruders, spammers, attackers [3]. Anonymous services have emerged that can filter out abusive and harmful messages. Most social networks are currently outsourcing this difficult task to external surveillance companies, some of which embrace crowdsourcing solutions. In contrast, *Rayzit* uses an in-situ crowdsourcing mechanism to remove abusive messages.

C. Conversational Commerce

There is a shift towards using social networks, messaging platforms and even simple SMS to deliver products to customers, who prefer expressing their needs by texting instead of looking for online vendors and following the classical online retail procedure. Magic[10] and Scratch[11], allow users to

[4] The Crowd Maps Beijing Floods: http://goo.gl/0DHZ4v

[5] WorkingWithCrowds: http://www.workingwithcrowds.com/

[6] Love Parade disaster: http://goo.gl/2FpIbm

[7] Hillsborough disaster: http://goo.gl/xvLRlc

[8] Opphos: http://www.sics.se/projects/opphos

[9] CrowdControlGames: http://crowdcontrolgames.com/

[10] Magic: http://getmagicnow.com/

[11] Scratch: http://www.tryscratch.com/

register to an SMS service and ask for anything they need. The applications personnel figures out a way and a price to deliver, and after confirming with the requestor, the delivery plan is implemented. The application Nativeapp.com is offering advanced travel agency services through messaging. The instant messaging app imQQ.com has been offering shopping services to its users since 2007 in China using virtual money. Path Talk[12] allows its users to message businesses directly to order goods.

Online VoIP and messaging applications, such as Skype.com, Viber.com, Tango.me, Kakao.com, LINE.me, WeChat.com, Facebook Messenger, and even SnapChat already sell digital products and are moving towards the direction of general e-commerce[13].

D. AkNN computation

The *k Nearest Neighbors (kNN)* of a user u from some dataset U, denoted as *kNN*(u, U), are the k users that have the most similar attributes to u [15]. kNN search is a classical Computer Science problem with many centralized algorithms that find applications in computational geometry [5], [4], [9], image processing [12], spatial databases [7], [24], [14], and social networks [1]. An *All kNN (AkNN)* query, viewed as a generalization of the basic kNN query, computes the *kNN*(u, U) result for every $u \in U$ and has a quadratic worst-case bound.

There are several AkNN algorithms optimized for offline analytics, aimed at *memory-resident data* [6], [4], [5], [9], [17], but also *disk-resident data* [7], [20], [23], [24]. Moreover, there are previous works that were utilizing the kNN properties in order to extend a social network to gain accuracy and predictability [13].

E. Big data and data mining

Social networks have billions of users generating petabytes of data every day. Additionally, the growth of smartphones, wearables and portable devices add new information such as location, biometrics values and environmental metrics increasing the internet traffic. In most cases the multidimensional data values consist of the user's location and the profile of the user as described by Jensen et al [11].

As a result, Big data infrastructures are necessary in order to accommodate the huge volume of the multidimensional data. Moreover, it is common that data is communicated and stored using semistructured formats such as JSON, XML etc. Consequently, a high performance NoSQL database management system is typically used to store, edit and manipulate the aforementioned data.

III. The *Rayzit* Architecture

In this section, we describe our *Rayzit* crowd messaging architecture. The architecture consists of various components that provide the required functionalities for the end-user.

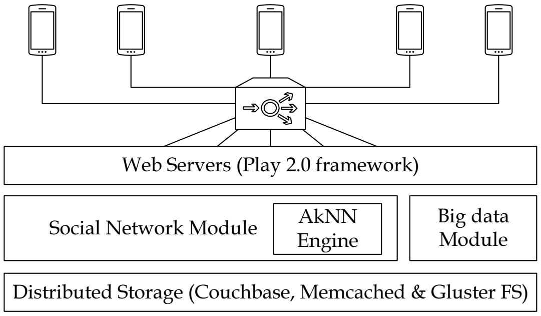

Web Servers (Play 2.0 framework)
Social Network Module
AkNN Engine
Big data Module
Distributed Storage (Couchbase, Memcached & Gluster FS)

Fig. 2. *Rayzit* distributed architecture

A. Architecture

The *Rayzit* architecture comprises of several modules utilizing state-of-the-art technology. It features a HAProxy[14] HTTP load balancer to distribute the workload to respective WWW servers (see Figure 2). Each WWW features a Play 2.0[15] server and a Couchbase NoSQL document store[16] for storing the messages posted by our users. In Couchbase, data is stored across the cluster in JSON format, which is indexed and directly exposed to the *Rayzit* Web 2.0 API[17]. In the backend, we also run the computing cluster that carries out the AkNN computation as we have already discussed in Section II. The results are passed to the WWW servers through main memory (i.e., Memcached) every few seconds.

B. Load balancer and web servers

Rayzit has multiple web servers in order to support a large amount of requests at the same time. In order to achieve this, we use HAproxy, a lightweight load balancer, to distribute the load to each web server. The web servers are deployed using the Play 2.0 framework, which is stateless, scalable, fast and thus can absorb large numbers of requests. In addition, the web servers host the Faye[18] system in order to serve a live feed for the end-user. Faye is a publish-subscribe messaging system based on the Bayeux protocol.

C. Social network module

The architecture can incorporate any application that wants to use the kNN network. In our case we have implemented and incorporated our *Rayzit* messaging app. The users can broadcast their messages to the k closest users or forward messages they see in order to propagate them to their own k closest neighbors. The proposed architecture maintains a core engine that generates the needed AkNN links in a distributed manner described next.

D. AkNN engine

Finding the k geographically nearest neighbors of all users is known as the *All k Nearest Neighbors* (AkNN) query. AkNN

[12] Path Talk: `http://path.com/talk`
[13] Forbes: `http://goo.gl/s0gYEM`
[14] HAProxy: `http://haproxy.1wt.eu/`
[15] Play: `http://www.playframework.com/`
[16] Couchbase: `http://www.couchbase.com/`
[17] *Rayzit* API: `https://api.rayzit.com/`
[18] Faye: `http://faye.jcoglan.com/`

query processing algorithms are used in several offline analytic queries. Recent techniques have shown how to scale such queries to shared-nothing cloud architectures using algorithms implemented in MapReduce. Yet, these algorithms are very slow for online operational queries, as needed by *Rayzit*, even when operating in distributed main-memory (e.g., Tachyon as opposed to HDFS).

Rayzit mandates AkNN query executions every few seconds, so that users can exchange microblog messages with their k geographically closest neighbors in an online manner. For this reason, we have developed a fast AkNN algorithm in our previous work [6]. This implementation makes use of a fast partitioning algorithm that achieves good load balancing, and a technique to identify a minimal set of points to exchange between neighboring partitions in order to compute the correct answer to the AkNN query in a distributed and parallel fashion.

E. Distributed storage

Rayzit architecture was built on a Big data NoSQL document-oriented database that is optimized for intensive applications. Couchbase was chosen in order to maximize the scalability and the performance of our system. We achieve a low latency access to the high volume of documents due to the Memcached layer [21]. Note, that our *Rayzit* app also allows users to attach files in their posts (i.e., photos, audio and videos). Therefore, a distributed file system needs to complement our proposed architecture in order to handle these media files. We use the Gluster[19] file system in order to secure the capability of scaling up to several petabytes and maintain a 24/7 service for thousands of clients.

F. Big data management module

Rayzit has thousands of users and it is necessary to monitor the users' behavior considering offensive, bullying and inappropriate messages. Consequently, we construct a Big data module in order to feed the administrator monitor panel with the required information.

Each message can be reported by any user in the crowd and the report is stored in our NoSQL database. We adopt a crowdsourcing majority vote technique to reach a decision. A single query extracts the reported rayz messages from the database, which consequently can be removed.

IV. Rayzit Application

Rayzit was launched in 2014 and has redefined anonymous messaging and social interactions. The main functionality of an anonymous social network is to provide ways for users to publish opinions, thoughts or gossip while preserving anonymity. Connecting each user with the k geographical closest peers (strangers) allows them to communicate anonymously. These connections are updated as users move.

The *Rayzit* mobile application allows users to send messages (rayzes) with media attachments, and reply to rayz posts based on a kNN network. In more details, the messages will be sent to the k nearest users. For each rayz that a user sends, the power of the user is decreasing (see Figure 3 (right)). When the power of a user reaches 0 they can not post or reply any more (see Figure 3). This functionality was incorporated in order to minimize spamming. The power of the user is fully reset every 24 hours or if the user's rayz is powered up by another user. In addition, a user can star or re-rayz any rayz which they consider interesting. Re-rayzing is the procedure of re-posting an existing rayz to one's own k nearest peers (similar to the re-tweet functionality in Twitter) and therefore propagating the post to more users.

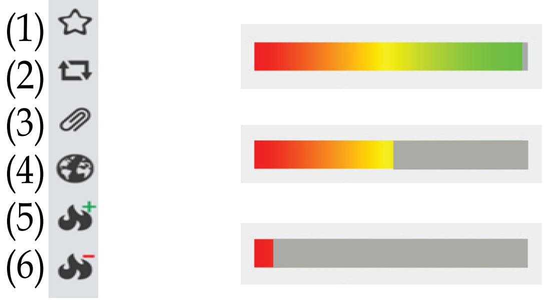


Fig. 3. (Left) The icons in *Rayzit*: A user can "star" a post (1), "re-rayz" a post to their k closest peers (2), attach a file to new post (3), setting a distance cut-off parameter if needed (4), "power up" (5) or "power down" (6) a post and its author. (Right) The anti-spam power bar: the power of each user decreases as they post, and increases every 24 hours or every time one of their posts is "powered up".

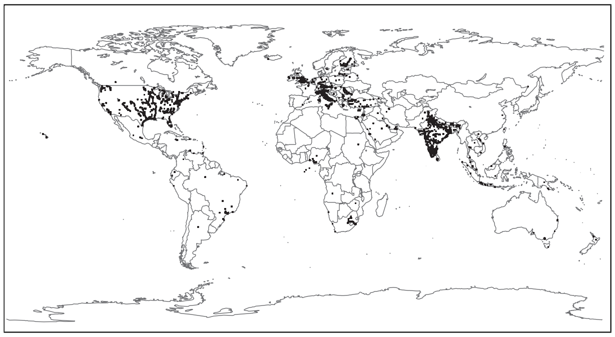


Fig. 4. *Rayzit* global user community.

Figure 4 shows the collection of active users in *Rayzit*. It is evident that most users have their nearest neighbors inside a small area. Now consider the scenario, where somebody has lost their dog and want to start looking in the neighborhood in order to find it and at the same time ask for help from the social network. *Rayzit* offers this functionality out of the box. The user can post a picture of the dog, which the users in the vicinity will see. If someone sees a dog that matches the attached picture (see in Figure 1 (right)), they can reply to the post with whereabouts and/or photos of the lost pet.

In order to comprehend how the *Rayzit* architecture works, consider the aforementioned scenario in terms of communication interactions. First of all, the client sends the rayz attaching a photo of the lost dog. The post, the photo, and their location is sent to the load balancer, which assigns them to a web server. The responsible web server stores the rayz through the Big data module. Then the social network module uses the dynamic social AkNN network computed by the AkNN engine in order to propagate the rayz to the k nearest neighbors. The

[19] Gluster: `http://www.gluster.org/`

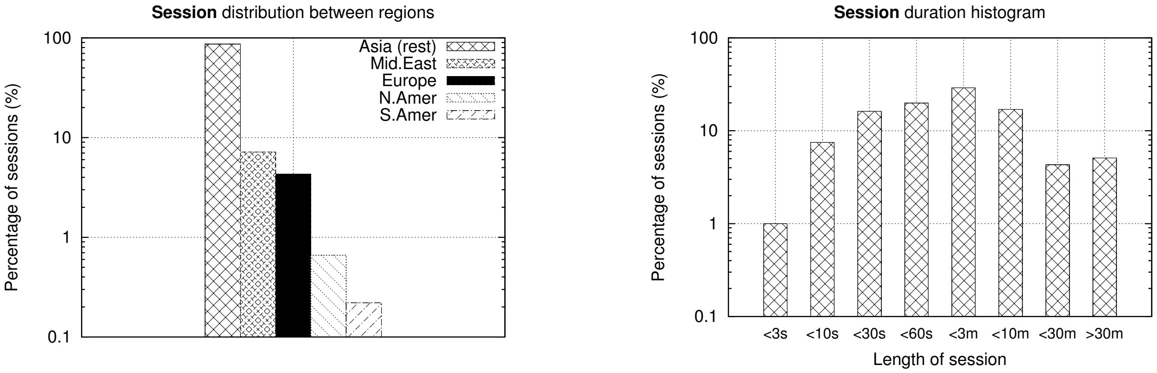


Fig. 5. (Left) Distribution of *Rayzit* users across regions in log scale. (Right) Average duration of each user session (reading and/or posting) in log scale.

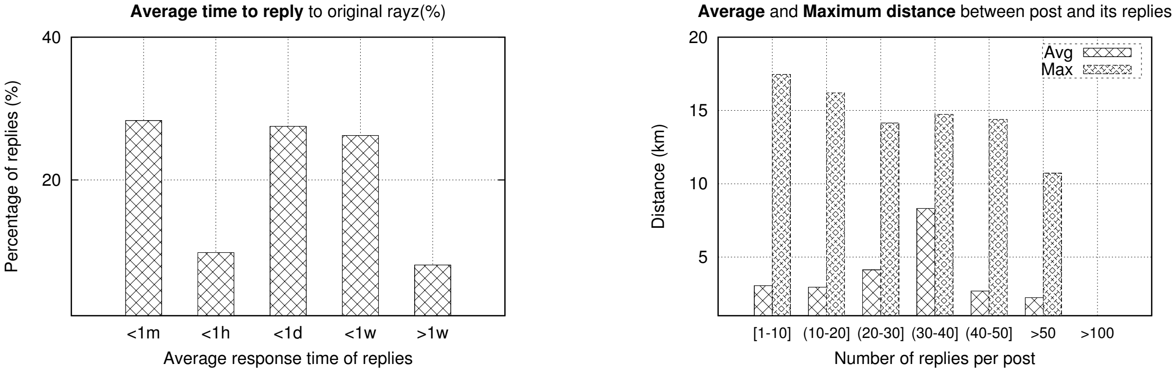


Fig. 6. (Left) Average time elapse between a post (rayz) and all its replies. (Right) The average and maximum distance between the locations of a post (rayz) and its replies, grouped by number of replies per post.

k nearest neighbors will receive the rayz into their live feed and they can reply if they have any information.

V. Data analysis and Evaluation

In this section, we have collected data produced by our *Rayzit* application and provide a brief data analysis. The data consists of several elements such as UUID and timestamp. Furthermore, we conduct experiments for evaluating and analyzing the performance of *Rayzit*.

Our evaluation focuses on two aspects: (1) the average percentage of replies per rayz at a specific time; and (2) how the distance affects the number of replies. We used Flurry.com analytics to get usage and distribution statistics of *Rayzit* app.

A. Session measurements

In this experiment we study the usage of *Rayzit* app in respect to region and duration of interaction.

The histograms in Figure 5 show the distribution of sessions in percentage. The session length is defined simply as the length of time between the start application event and the end application event. The Figure 5 (right) plots the distribution of the sessions by displaying the number of sessions for which the session length falls into predefined duration slots.

Figure 5 (left) plots the distribution of sessions in percentage for each region. More than 87.6% of sessions are created in Asia, and more than 7.2% of them in Middle East. 4.3% of all sessions are created in Europe. Only 0.7% of sessions are created in North America. Only 0.2% of the sessions have been created in South America.

Figure 5 (right) plots the distribution of *Rayzit*'s sessions across duration time slots. It is noticeable that 1% of the sessions is completed within 3 seconds. 7.50% completed between 3 and 10 seconds and 16.20% observed between 10 and 30 seconds. 19% of the sessions completed between 30 and 60 seconds. The time slot of 1 to 3 minutes has the maximum percentage with 29%. Furthermore, 17% completed within 10 minutes and 4.3% between 10 and 30 minutes. Finally, only the 4.3% completed after 30 minutes. Consequently, the AkNN engine needs to deliver the dynamic kNN graph in less than a minute to satisfy most users.

B. Interaction measurements

In this experiment we study how users interact through *Rayzit*, and in particular how fast a user replies to a post and how the distance affects interaction.

We used real data that was collected through the *Rayzit* back-end. The data include the time and location stamp of a post or a reply. The time between a rayz and a reply is presented in time slots: $(< 1m)$ less than a minute, $(< 1h)$ between a minute and an hour, $(< 1d)$ between an hour and a day, $(< 1w)$ between a day and a week and $(> w)$ more than a week. The average and maximum distances of the replies for each rayz were measured in order to depict how the distance affects the number of replies.

Figure 6 (left) plots the distribution of the reply arrival time, which is the time between each reply and the original rayz. 28% of replies arrive within the first minute of the original rayz, and more than 28% of replies arrive within a day. Only 10% of replies arrive with in the first hour. 26% of replies arrive between the first day and the first week. Only the 8% of replies arrive one week or more after the rayz creation. This confirms our intuition for the temporal dimension of a rayz: it is very unlikely to get attention after a while.

Figure 6 (right) plots the distance that a rayz can reach. It is noticeable that rayz posts with 1 - 50 replies have a maximum distance of almost 15km. Rayz posts with more than 50 replies have 10km maximum distance and 2km average distance. Moreover, rayz posts with more than 100 replies have only 8m maximum and 6m average distance. This validates our initial claim that the nearest users are more likely to be related to a specific subject than the faraway users.

VI. Conclusion

This paper presents an innovative architecture for anonymous dynamic social networks, coined Rayzit, which enables anonymous interactions with the k nearest neighbors using state-of-the-art technology. An industrial quality application was implemented that utilizes the aforementioned architecture. We presented and analyze the data collected from this application and draw interesting conclusions about this new type of dynamic social networking. Our experimental results also confirm our initial hypothesis that the number of the replies is related to the location of the original rayz (i.e., the rayz with the most replies has a maximum range of 8 meters).

As future work, we will investigate further the advantages of the dynamic AkNN graph in order to enhance user experience in our social application. In addition, better content filtering mechanism will be studied in order to provide the best possible quality to the users.

Acknowledgment

This work was supported by an *Appcampus Award* by Microsoft, Nokia and Aalto University (Finland). It has also been supported by the University of Cyprus (Startup Grant / Zeinalipour), by MTN Cyprus, EUs COST Action IC903 and IC1304 and EUs FP7 MODAP project.

References

[1] N. Armenatzoglou, S. Papadopoulos and D. Papadias. "A general framework for geo-social query processing". *In Proceedings of the 39th international conference on Very Large Data Bases (PVLDB'13)*, 913–924, 2013.

[2] M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, and M. Tesconi, "EARS (earthquake alert and report system): a real time decision support system for earthquake crisis management". *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14)*, 2014.

[3] A. Bozzon, M. Brambilla, S. Ceri. "Answering search queries with crowdsearcher". *In Proceedings of the 21st ACM international conference on World Wide Web. WWW '12*, 1009–1018, 2012.

[4] P.B. Callahan. "Optimal parallel all-nearest-neighbors using the well-separated pair decomposition". *In Proceedings of the 34th IEEE Annual Foundations of Computer Science (SFCS '93)*, 332–340, 1993.

[5] K.L. Clarkson. "Fast algorithms for the all nearest neighbors problem". *In Proceedings 24th Annual Symposium on Foundations of Computer Science 1983 (FOCS'83)*, 226–232, 1983.

[6] G. Chatzimilioudis, D. Zeinalipour-Yazti, W.-C. Lee and M.D. Dikaiakos. "Continuous all k-nearest neighbor querying in smartphone networks". *In Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM'12)*, 79–88, 2012.

[7] Y. Chen and J.M. Patel. "Efficient evaluation of all-nearest-neighbor queries". *In Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE'07)*, 1056–1065, 2007.

[8] A. Doan, R. Ramakrishnan, A. Y. Halevy, "Crowdsourcing systems on the world-wide web". *Communications ACM 54*, 86–96, 2011.

[9] H.N. Gabow, J.L. Bentley and R.E. Tarjan. "Scaling and related techniques for geometry problems". *In Proceedings of the 16th ACM symposium on Theory of computing (STOC'84)*, 135–143, 1984.

[10] H. Gao, J. Tang, X. Hu, and H. Liu, "Modeling temporal effects of human mobile behavior on location-based social networks". *In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM '13)*, 1673–1678, 2013.

[11] C.S. Jensen, A. Kligys, T.B. Pedersen, and I. Timko, "Multidimensional data modeling for location-based services". *The VLDB Journal 13*, 1–2, 2004.

[12] T.H. Lai and M.-J. Sheng. "Constructing euclidean minimum spanning trees and all nearest neighbors on reconfigurable meshes." *IEEE Transactions of Parallel Distributed Systems*, 7, 806–817, 1996.

[13] N. Lathia, S. Hailes, and L. Capra, "kNN CF: a temporal social network". *In Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08)*, 227–234, 2008.

[14] M. Renz, N. Mamoulis, T. Emrich, Y. Tang, R. Cheng, A. Zufle, and P. Zhang. "Voronoi-based nearest neighbor search for multi-dimensional uncertain databases", *In Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE '13)*, 158–169, 2013.

[15] N. Roussopoulos, S. Kelley and F. Vincent. "Nearest neighbor queries". *In Proceedings of the ACM SIGMOD international conference on Management of data (SIGMOD '95)*, 71–79, 1995.

[16] T. Steiner, "DC proposal: enriching unstructured media content about events to enable semi-automated summaries, compilations, and improved search by leveraging social networks". *In Proceedings of the 10th international conference on The semantic web - Volume Part II (ISWC'11)*, Vol. Part II. Springer-Verlag, 365–372, 2011.

[17] P.M. Vaidya. "An o(n log n) algorithm for the all-nearest-neighbors problem". *Discrete Computational Geometry*, 4, 101–115, 1989.

[18] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B.Y. Zhao, "Whispers in the dark: analysis of an anonymous social network". *In Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*, 137–150, 2014.

[19] M. Watanabe and T. Suzumura, "How social network is evolving: a preliminary study on billion-scale twitter network". *In Proceedings of the 22nd international conference on World Wide Web companion (WWW '13 Companion)*, 531–534, 2013.

[20] C. Xia, H. Lu, B. Chin Ooi and J. Hu. "Gorder: an efficient method for knn join processing". *In Proceedings of the 30th international conference on Very large data bases (VLDB '04)*, 756–767, 2004.

[21] C. Xu, X. Huang, N. Wu, P. Xu, and G. Yang. "Using Memcached to Promote Read Throughput in Massive Small-File Storage System". *In Proceedings of the 9th IEEE International Conference on Grid and Cloud Computing (GCC '10)*, 24–29, 2010.

[22] D. Yang, D. Zhang, K. Frank, P. Robertson, E. Jennings, M. Roddy, and M. Lichtenstern, "Providing Real-time Assistance in Disaster Relief by Leveraging Crowdsourcing Power". *Personal Ubiquitous Computing 18, 8*, 2025–2034, 2014.

[23] B. Yao, F. Li and P. Kumar. "K nearest neighbor queries and knn-joins in large relational databases (almost) for free". *In Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE '10)*, 4–15, 2010.

[24] J. Zhang, N. Mamoulis, D. Papadias and Y. Tao. "All-nearest-neighbors queries in spatial databases". *In Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM '04)*, 297–306, 2004.