Plans and Patterns: Network Performance Testing and Data Transformation for POSN

Patrick Austin
Department of Computer Science and Engineering
University of Nevada, Reno
paustin@nevada.unr.edu

***Abstract*—Online Social Networks (OSNs) have become some of the most popular platforms for internet communication. Most popular OSNs are designed around a centralized client-server architecture, where user data is stored and handled by a single managing authority. When that authority's security is compromised, there is potential for vast breaches of user privacy. In order to advance research towards the possibility of a practical decentralized OSN architecture where privacy and security of user data can be emphasized to a higher degree, we have developed on existing work on POSN, a prototype cloud-supported decentralized OSN. In order to achieve a better understanding of network conditions in which POSN might operate, we have created a simple Android platform for testing performance of cloud and OSN providers in various environments for various types of data. We have also conducted extensive evaluation and transformation of previously gathered data about patterns of OSN use on Facebook in order to extract useful information and conclusions about user behavior and demands on centralized OSNs, which helps us to evaluate the feasibility of a cloud-supported decentralized OSNs.**

I. Introduction

Online social networks (OSNs) such as Facebook have become some of the most widely used internet applications [1]. The vast majority of widely used OSNs are based on a centralized client-server architecture where user data and user authentication are stored and handled by a single authority which administers the OSN. However, centralized architectures can raise privacy issues. Users interested in seeing their personal data protected against large-scale security breaches, sale to advertisers, and exploitation by marketing and analytics departments may desire OSNs that place a higher priority on the user's privacy and control over their own data. Therefore it may be desirable and important to experiment with distributed OSN architectures based on a peer-peer model, where user data is never stored on a centralized server at all. There may be an underserved market niche in the OSN space that could be satisfied by catering to users who place a high priority on privacy and data control.

However, there are a number of challenges which must be solved in order to realize a practical, usable decentralized OSN. Unlike in a centralized model, where the server is assumed to have very high availability, the peers in a decentralized model can go online and offline intermittently, potentially impeding access to data that needs to be distributed to other OSN users. While centralized servers can use specialized hardware and high-bandwidth internet connections, a peer-peer architecture will have to leverage consumer hardware, primarily mobile devices, as well as consumer internet connections, which are likely to have relatively low bandwidth ceilings and monthly data caps. Additionally, authentication and the exchange of data is a more complicated problem when friends in the OSN must communicate directly instead of via the server, which can serve as middle-man and handle problems of authentication and storage of data.

Achieving a high degree of optimized performance and scalability is one of the missing pieces that must be realized to create a peer-peer OSN that could realistically be used by a mass audience. Previous attempts in this area have involved varying degrees of compromise, including experimentation with hybrid approaches that combine aspects of the client-server and peer-peer models, but it would be ideal to avoid exposing data to a server or the general public as much as possible. There are many possible approaches to implementation of even basic OSN features such as comment propagation in a distributed environment; finding the most optimal solution to match real-world user behavior is non-obvious. The best ways in which to utilize and leverage varying types of consumer devices and resources (mobile devices, personal computers, IoT devices, access to cloud computing, etc.) within the distributed OSN is also an unsolved question which merits further research.

Two pressing areas of research for designing a practical peer-peer OSN are delegated comment propagation and the incorporation of heterogenous devices. Delegated comment propagation involves the decision by a user who is going offline to delegate management of comments on the user's posts to one of their friends until they return. That user can then delegate control to another mutual friend when he goes offline, and so forth. Several possible approaches for selecting the delegate can be considered and compared via simulation in terms of performance. By incorporating a type of heterogenous device, user PCs, into the distributed OSN, we can explore the extent to which introducing high-availability devices with larger storage and network capacity can improve performance.

In this paper we attempt to advance the research into these and other open questions. First, we have created a basic platform for conducting network performance testing for OSN and cloud providers. Second, we have engaged in an extensive process of data transformation and evaluation on previously gathered OSN and cloud network performance data to compare against the needs of a prototype peer-peer OSN. This work builds on prior work done on and data gathered for the

POSN distributed social network prototype project created at the University of Nevada [2][3][4], Reno. The data used in this simulation was gathered from real-world Facebook users in a prior study at the University of Nevada, Reno, and should provide realistic real-world behavior against which the viability of POSN's cloud oriented approach can be evaluated [5].

II. RELATED WORK

There have been several significant prior attempts to create or prototype distributed OSNs, each offering a different approach and different insights and compromises that might be used to address the challenges inherent to a distributed OSN structure.

The most highly relevant prior attempt for this paper is POSN, created at the University of Nevada, Reno [2][3][4]. Much of the work we conduct in this paper explores and builds on the POSN architecture by simulating and evaluating the addition of several new features. POSN is centrally motivated by the desire to leverage powerful, widely available, and low-cost cloud computing resources such as Google Drive, OneDrive and Dropbox to achieve a secure, high-availability secondary location from which a user's OSN activities can be managed besides their primary mobile device. This helps address the availability problem caused by a user going offline intermittently in the peer-peer architecture, and encrypting the OSN content stored in the cloud maintains the emphasis on security and privacy that motivates most distributed OSNs. The authors of POSN experiment with several cloud computing providers as potential supporting platforms and detail schemes for supporting data structures and feature implementations. Their work is supported by analysis and simulations conducted using a dataset of real-world Facebook user behavior [5].

While we find the initial results of POSN compelling, the prospect of additional refinement is also appealing. In a distributed OSN architecture there is significant pressure to achieve highly optimized network performance so as not to place an undue burden on users. An OSN that consumes a large part of a user's monthly data cap distributing someone else's posts is unlikely to achieve widespread popularity. Likewise, it remains to be demonstrated beyond doubt that cloud platforms are capable of supporting an OSN environment in terms of performance demands; this is one question that we will explore in this paper.

The assumptions made in the paper are generally reasonable. However, some further economic and practical questions include whether the public demand for a distributed OSN is high enough to support attempts to commercialize a product like POSN. That is, how much does the general public care about privacy versus the convenience and popularity of existing platforms? How is such a platform to monetize when it cannot tap user data like existing platforms? Also, there can be some question about the viability of cloud computing as a supporting platform were POSN to be adopted on a large scale beyond initial questions of feasibility. If POSN achieved high enough data usage to pass the thresholds at which cloud computing platforms require user payment, one can wonder whether users would pay when free options for OSNs are available. One can also wonder whether organizations like Google would oppose their cloud computing platforms being used to support a rival social network.

Another distributed OSN to consider is Supernova [6]. Supernova is particularly relevant due to its conception of a super-peer. A super-peer is a user in the distributed OSN who has additional perks as well as responsibilities within the OSN. Essentially, users can earn the right to be a super-peer by providing storage, computation, and bandwidth for other nodes in the network. One incentive the authors propose to encourage users to become super-peers is the ability to serve advertising to normal users the super-peer supports.

Supernova is particularly relevant to this study because it proposes leveraging the heterogeneity of devices in the OSN and tries to exploit differences in hardware and behavior between users to achieve optimal OSN performance. This observation is similar to the one motivating our desire to simulate the introduction of PCs into the architecture of POSN, and some lessons of Supernova may be transferable. For one, a major result the authors observe is that the dependence on super-peers can create widespread availability issues when the super-peers go offline; even though super-peers have higher availability than normal users, the super-users can still go offline at unpredictable times. In other words, one loses some of the benefits one would expect of a distributed network where no user is dependent on a single server-like device. This may also prove to be the case if PCs are expected to support large numbers of users in the POSN simulation. The authors also briefly investigate the notion of incentive to serve as a super-peer, which is an important question for a real-world implementation of the PC-supported POSN beyond the scope of this paper. However, the authors do not focus on the security challenges facing a super-peer scheme where one user is handling the data of many others, whereas POSN is built from the ground up to address such security concerns and maintain the privacy of user data.

Another distributed OSN that may be relevant is Diaspora. Unlike many of the theoretical or prototyped distributed OSNs found elsewhere in the literature, Diaspora has been fully implemented and released as open source software, boasting over 600,000 users [7][8]. Diaspora has major differences from POSN in terms of structure, essentially allowing independently owned and operated servers, called pods, that interact to create the larger network. This hybrid approach is not very similar to the peer-peer emphasis of POSN, but some lessons may be transferrable. Interestingly for future research, Diaspora can be used as a platform for further prototyping of OSNs [9].

The real-world nature of Diaspora also offers lessons about some of the potential problems of OSNs in actual operation. In one concerning episode, members of ISIS began to use Diaspora as a communications platform in 2014, and due to the decentralized nature of the network the operators of Diaspora could not actually remove the content without the consent and cooperation of the relevant pod administrators

[10]. This suggests just a few of the legal and political challenges distributed OSNs can face as they emerge from the realm of theory into the realm of practice.

III. Methodology

We initially set out to achieve several goals related to building on, advancing, and improving prior work done on the POSN platform. Broadly, these goals fell into categories of network testing, data transformation and visualization, and feature simulation.

In terms of network testing, we set out to create a platform that could collect up-to-date and fresh data about the performance of real world OSNs and cloud computing platforms in real-world use cases to compare against POSN. In terms of data transformation, we set out to conduct further analysis of the previously gathered Facebook user data [5] to extract useful data and visualizations that show the potential viability of POSN's cloud-oriented approach. In terms of feature simulation, we set out to expand prior work done on POSN simulation to incorporate and evaluate comment propagation strategies and the introduction of heterogenous devices in the form of user PCs.

A. Network Testing

In order to have an up-to-date metric to compare and evaluate POSN performance against, we first set out to utilize the dataset of real-world Facebook user behavior previously collected at the University of Nevada, Reno [5] to do network testing. By collecting data about the network performance of existing OSN and cloud computing platforms, quantifiably evaluating the performance of the POSN platform and the viability of its architecture at scale is more easily achievable.

Our approach began with transforming the existing dataset, which consisted of details about the behavior and activity of the friend circles of 16 Facebook users who participated in the study over 15 days in 2014. We focused on multimedia, which poses a larger challenge in terms of bandwidth in OSN communication compared to text messages such as comments or private messages due to the far larger file sizes. We aggregated the image and video files disseminated in these friend groups over the course of the study and sorted them into a single set of file size values. Figure 1 shows a graph of the resulting dataset sorted in descending order of size.

Overall, about 16GB of multimedia data was observed in the original experiment; approximately 12,000 multimedia files were disseminated over the friend groups of all 16 users during the course of the study. These multimedia file sizes give some perspective on the performance demands facing a modern OSN. The majority of the dataset consists of small image files between about 10KB and 100KB in size. There are fewer but much larger video files, with a few instances exceeding 100MB in size. The goal of our network testing was to quantify how existing centralized OSNs, along with some of the prospective types of cloud platforms that could be used to support POSN, handle files with these size characteristics.

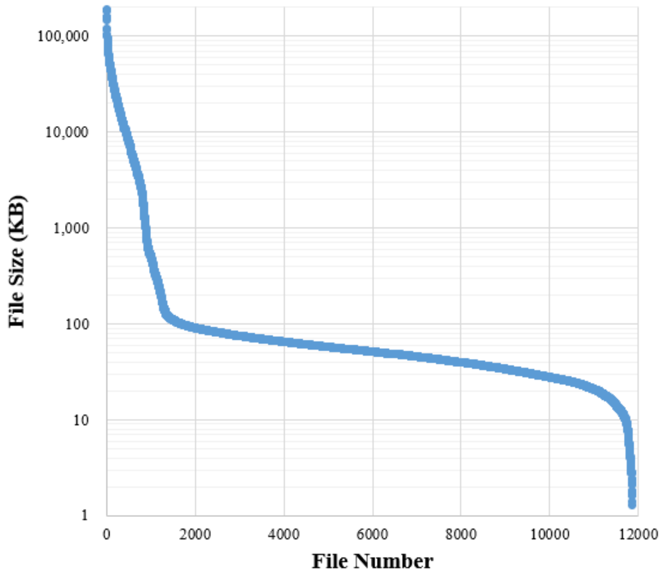

Fig. 1. File sizes of the aggregated Facebook multimedia data (log scale)

Since Facebook is one of the largest players in the OSN space [11] and a prominent user of the centralized OSN architecture, we particularly emphasized the collection of Facebook performance data as applied to this dataset. Fortunately, the Facebook API supports the creation of bots that can communicate over the Facebook Messenger platform [12]. Since the Facebook Messenger platform supports direct transmission of arbitrarily sized files such as PDFs between friends, we approached the problem by creating two Facebook bots that would send and receive files of the requisite sizes between themselves. We could then evaluate upload and download speeds for the file transmissions.

Likewise, the app could repeat these file transmissions on cloud platforms such as Google Drive, OneDrive, and Box in a more straightforward fashion, simply by uploading and downloading appropriately sized files to and from cloud storage. This approach would allow for the evaluation of cloud performance on data fitting the profile of use shown in the Facebook dataset.

In order to create as pure of a network environment as possible on the local device, where there would be no interfering processes communicating on the network, using bandwidth, and possibly influencing our results, we created a simple Android application running on a stripped-down device which would generate appropriately sized files and transmit them via the relevant APIs while collecting data about the observed upload and download speeds.

Initial design and testing of the data collection application was to be carried out in Reno and tested on local consumer networks as well as the University of Nevada, Reno network. However, network latency and performance will vary from location to location, especially when there are no nearby data centers for the platforms being used. In order to collect more generalized data reflecting broader network performance across the United States and internationally, we planned to gather data via the app in several additional states and countries via the cooperation of individuals at those locations as well as via VPN.

B. Data Transformation and Visualization

Although a large amount of user data had been captured in the Facebook friend circles experiment [5], presenting and interpreting that data in an effective human-understandable way proved a considerable challenge in and of itself. In addition to the aforementioned aggregated multimedia file size data (Figure 1), we had access to extensive event logs of activity for each of the 16 Facebook users who participated in the original study, as well as prior network measurement diagnostic data gathered by the POSN researchers.

The event data that was primarily of interest to us documented how many multimedia posts had been made in the user's friend circle since the previous time they were online. Were we to transpose this behavior into the POSN environment, this would signify how many posts the POSN user would have to download from peers in order to be able to see all current content. The logic was to use this data as a kind of test bed to evaluate the kind of network behavior that would be expected in POSN, assuming POSN users behaved in a similar fashion to how they would on Facebook.

Of the prior network measurement data, we were primarily interested in upload and download throughput measurements taken for various cloud and OSN platforms for various file sizes. For a variety of reasons related, among other things, to TCP implementation and saturation of bandwidth, POSN's researchers had observed a highly non-linear relationship between the size of a file being downloaded and the throughput of the download and significant variance in behavior between OSN and cloud platforms. It was necessary to transform and interpret the data in order to evaluate the impact of these observations on POSN feasibility. That is to say, did the types of files being transmitted on OSN have size characteristics that meant they were poorly suited to cloud? Were cloud platforms likely to be able to handle POSN traffic in an efficient fashion? Would the network delays expected in average use of POSN be realistic and tolerable to users? We set out to evaluate these and similar concerns.

We used Microsoft Excel as the primary platform for this data exploration, supported by Python scripting with the Openpyxl library for more complicated transformation tasks, which we will discuss in more detail shortly.

C. Feature Simulation

While our primary focus was on achieving successful network testing and data transformation, as time allowed we also planned to explore the possibility of expanding prior POSN simulations to incorporate comment propagation strategies and one type of heterogenous device, user PCs. Our intended strategy was to simulate POSN's features as applied to the OSN behavior we observed in the Facebook data; the hope was that this behavior would map to realistic, average OSN user behavior.

Comment propagation is a necessary feature in POSN that had not been highly elaborated on or extensively implemented in prior work. In the POSN structure, the user who makes a post is responsible for storing, managing, and distributing the comments other users make on that post. However, since it should still be possible to see comments made on a user's posts when that user is offline, some kind of delegation strategy must be chosen to select a friend who is online to be the one who will manage the distribution of comments until the original owner returns. This delegation may in turn be handed off to another mutual friend when that original delegate goes offline, and so on.

Various strategies for selecting the delegate are possible. We intended to implement at least four basic strategies, involving delegation to (i) a random friend, (ii) the friend with the most common friends, (iii) the friend with the highest recent online time, or (iv) the friend with the highest disjoint online time. These strategies could then be evaluated on the OSN user behavior we collected from the Facebook dataset to see which exhibits the most efficient performance in terms of minimizing network connections.

In terms of heterogenous devices, we aimed to introduce PC devices into the simulation of the Facebook user behavior on the POSN platform. While POSN is already intended to be supported by highly available cloud platforms for the purposes of OSN data storage, having highly available users in the network is also a desirable property that would improve performance in the network. However, the extent to which performance would be improved by the introduction of these devices had not previously been tested via simulation.

These PCs would be introduced into friend networks as highly (though not completely) available devices with high storage capacity and bandwidth associated with a given POSN user. In theory, these PCs would be able to support the network to some extent when normal users on mobile devices went offline, assisting in activities such as the aforementioned comment propagation. The quantity of user PCs to be introduced into the networks would be a question of experimentation that could be varied in the simulation. As in the case of comment propagation, we would then attempt to evaluate the degree to which PCs make POSN communications more efficient, in terms of minimization of unnecessary connections and network traffic.

IV. Results

Due to limited time and resources available, and in order to best meet the needs of the researchers working on POSN with whom we collaborated, we conducted extensive work on the network testing and data transformation and visualization areas of the project, largely leaving the feature simulation area for future work.

A. Network Testing

We created a prototype standalone Android app called POSNTester in order to implement our network testing goals. POSNTester essentially attempts to upload and download a file of each size represented in Figure 1 to each of the OSN and cloud platforms of interest to the POSN researchers, including Facebook, Google Drive, OneDrive, and Box, gathering diagnostic data about upload and download time and throughput in the process and storing the results in a Google Sheet. This app could then be run in various networks and environments in order to collect data about cloud and OSN platform performance on varying file sizes.

POSNTester is largely powered by the CloudRail library, which offers a unified, uniform interface for interaction with cloud storage and messaging platforms [13][14]. CloudRail's high degree of convenience in abstracting away interaction with multiple highly different and often poorly-documented platform-specific APIs made it ideal for our purposes. Likewise, CloudRail abstracts away the process of changing the application when individual platforms update or change their APIs, which is highly convenient and useful for the long-term life of the POSNTester platform.

Since many of basic cloud platform interactions used in POSNTester are integral to the theoretical operation of POSN itself, and it would be desirable for POSN to operate with minimal need for long-term API maintenance by app developers, one of our major suggestions or takeaways from our results is that CloudRail may be highly useful in the implementation of POSN itself.

While the app was developed up to a working prototype phase, we opted not to progress to a robust final implementation and the collection of final data. For one thing, the large size of the multimedia file dataset would have meant that POSNTester would have had to upload and download 16GB of data for each platform being measured, which was likely to be prohibitively costly and difficult in terms of time and bandwidth and likely would have run up against limitations on data transfer imposed by the individual platforms. Additionally, the implementation of testing via Facebook Messenger proved to be more difficult and limited than expected, due to restrictions on supported file size (max 25MB) and the need for a Messenger bot to run on an independent server with a webhook. This would have greatly complicated the measurement process and may not have led to viable data for the Facebook platform, which was the platform of central importance to the POSN researchers. For these reasons we opted to stick with the previously measured data from the Facebook friend circles study.

B. Data Transformation and Visualization

After several initial experiments with the data and much discussion with the POSN authors about requirements, our focus turned to applying the previously measured throughput data, shown in Figure 2, to the available data gathered from the Facebook friend circle study about multimedia file traffic. This would allow the authors to visualize and explore download and upload times for files on OSN platforms versus cloud platforms; this would provide some tangible, human-understandable evidence of the viability or lack thereof of POSN for real-world use cases.

The primary issue that had to be addressed was the lack of continuous measurements for varying file sizes; determining how best to extrapolate throughput values for file sizes not explicitly present in Figure 2, namely much larger files, required extensive experimentation. Our first approach used a naïve bucketing strategy, where files were simply assigned the throughput value of the nearest data point. However, when visualized, this produced an unappealing and unrealistic sawtooth behavior which did not adequately represent the real-world behavior we were trying to capture, as shown in Figure 3. Bucketing was implemented via Python script with Openpyxl.

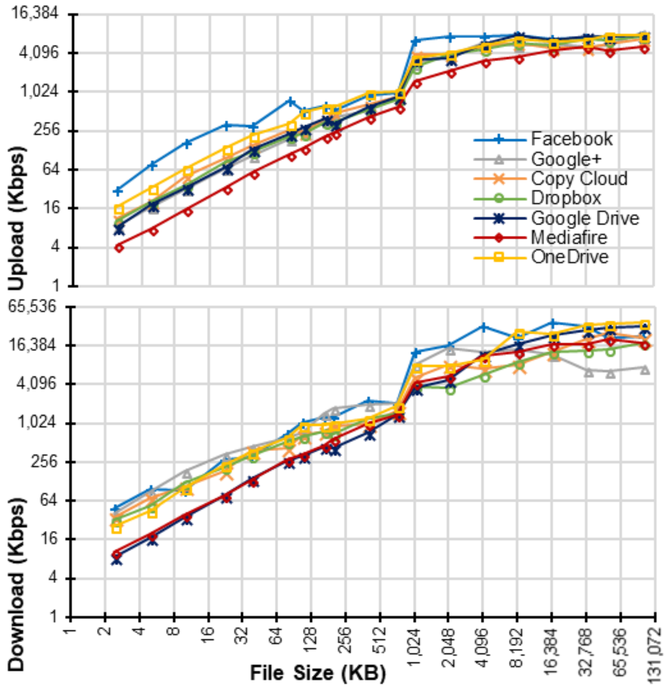

Fig. 2. Original throughput measurements for cloud and OSN providers in upload and download (log-log scale)

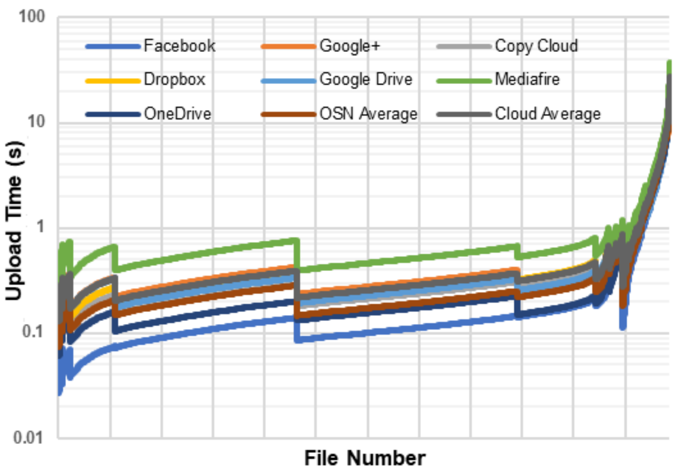

Fig. 3. Extrapolated upload times with bucketing; the sawtooth behavior observed was deemed unrealistic and unacceptable (log scale)

Since bucketing produced unacceptable results, we moved towards an approach that worked in a more continuous fashion by using several best-fit trendlines. As can be seen in Figure 2, the throughput values generally exhibit three phases of behavior. The first phase, for files less than 718 KB in size, is a power law relationship. The third phase, for files greater than 1024KB in size, is roughly logarithmic. Due to lack of captured data it was difficult to determine the relationship in the transition phase between 718 and 1024 KB; we assumed that range to exhibit linear growth. With these formulas determined for all platforms we could determine a throughput value for any arbitrary file size and use that to extrapolate upload and download times for given files. Our results for this continuous implementation approach are shown in Figure 4.

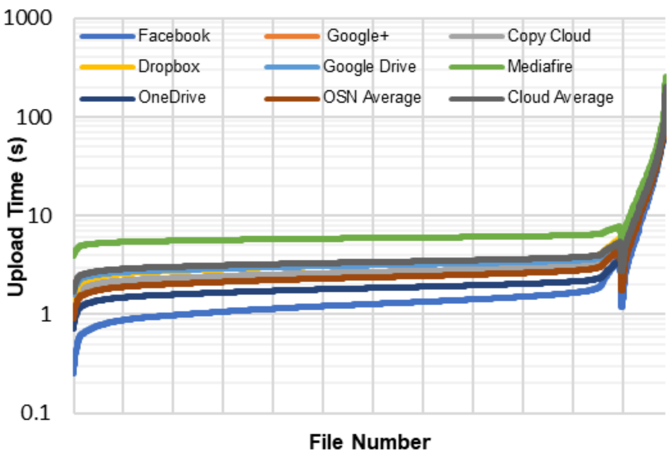

Fig. 4. Extrapolated upload times with trendline approach (log scale)

The trendline approach produced much more logical behavior, though it also significantly increased the upload time estimate for the few largest files in the dataset. In terms of results, this visualization helps us observe that generally the OSN and cloud providers are in a tight band of performance for the types of files observed in the Facebook user study, especially at the higher end, and both OSN and cloud platforms have very fast transmission times for the vast majority of files. The graphs also help us observe that among the platforms, Facebook and OneDrive generally achieve the best performance, i.e. the lowest upload time.

Additionally, the authors of POSN were interested in box plots to represent the five-number summary of the derived download and upload time statistics visually and on a per-user basis, to help compare Facebook and OneDrive performance as well as OSN and Cloud average performance generally. This involved additional scripting. An example of one the generated charts is shown in Figure 5. These graphs help us to conclude that OSN and cloud performance on the observed file sizes is similar on a per-user basis, with OneDrive generally exhibiting slower upload times in the first quartile but beating Facebook for the largest files.

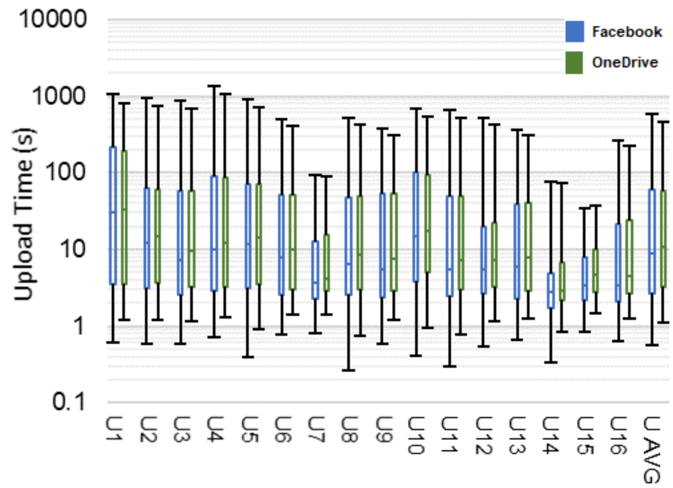

Fig. 5. Five-number summary comparison of Facebook and OneDrive upload times per user (log scale)

These are but a few of the graphs we were able to realize. All told, we were able to realize much of the data visualization desired by the POSN authors, which should assist them in explaining POSN concepts and making the case for the potential of the platform.

V. Future Work

While we were able to achieve much of what we set out to accomplish, some areas of study had to be left to future work.

A full implementation of POSNTester, if deemed necessary, would require additional testing and implementation of features, as well as careful examination of the Facebook case, where gathering good data may prove challenging. At the same time, CloudRail provides a good starting platform on which to begin, especially as applied to the cloud providers.

In terms of data transformation, most of our primary goals were achieved: namely, we gained several high-quality charts that help to visualize and explain POSN performance by showing not much difference in file upload and download behavior between cloud and OSN providers. Future work would likely involve the acquisition of new data or conducting additional simulation, then beginning the process of understanding, transforming, and visualizing the data anew.

The area where we were able to accomplish less was in terms of simulation of additional features like delegated comment propagation and heterogenous device support. Getting up to speed on the state of the POSN project and communicating as a distributed team proved time-consuming and put those objectives beyond the scope of this paper.

VI. Conclusions

As OSNs have grown, so has the severity of security breaches and the corresponding public backlash. Decentralized OSNs provide an alternative, competing vision for mass social interaction on the internet. But achieving a decentralized OSN that is secure and scalable to mass use which also supports all features expected of a modern OSN requires careful planning and thorough analysis of data. In this paper we have explored POSN's architecture at a high level and collected and analyzed data concerning OSN use and platform behavior, representing just a few of these plans and patterns. Our results give reason to believe that decentralized OSNs in general and POSN in particular may be nearly ready for a mass audience.

References

[1] "Facebook," Feb 2018, https://newsroom.fb.com/company-info/

[2] E. Erdin, E. Klukovich, M.H. Gunes, and G. Gunduz. "POSN: A Personal Online Social Network," in IFIP SEC, 2015.

[3] E. Erdin, E. Klukovich, and M.H. Gunes. "Cloud Assisted Personal Online Social Network," in Social Network Analysis Lecture Notes Series, Springer, 2018.

[4] E. Erdin and M.H. Gunes. "Efficient Data Dissemination in Distributed Social Networks," in Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2017.

[5] E. Erdin, E. Klukovich, and M.H. Gunes. "An Analysis of Friend Circles of Facebook Users," in Local Computer Networks Conference Workshops (LCN Workshops), IEEE 40th, 2015.

[6] R. Sharma and A. Datta, “Supernova: Super-peers Based Architecture for Decentralized Online Social Networks,” in Proceedings of the COMSNETS, Bangalore, India, 2012.

[7] “The diaspora* Project,” Feb 2018, https://diasporafoundation.org/

[8] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang, “The Growth of Diaspora - A Decentralized Online Social Network in the Wild,” in Proceedings of the IEEE INFOCOM Workshops, 2012.

[9] R. Gay, J. Hu, H. Mantel, and S. Mazaheri, “Relationship-Based Access Control for Resharing in Decentralized Online Social Networks,” in International Symposium on Foundations and Practice of Security, 2017.

[10] “Islamic State Fighters on diaspora*”, Feb 2018, https://blog.diasporafoundation.org/4-islamic-state-fighters-on-diaspora

[11] “Social Media Use in 2018”, March 2018, http://www.pewinternet.org/2018/03/01/social-media-use-in-2018/

[12] “Facebook Messenger Platform”, March 2018, https://developers.facebook.com/products/messenger/overview/

[13] “Cloudrail Unified Cloud Storage API”, May 2018, https://cloudrail.com/integrations/interfaces/CloudStorage

[14] “Cloudrail Unified Messaging API”, May 2018, https://cloudrail.com/integrations/interfaces/Messaging