

Estimating user interaction strength in distributed online networks

Jia, Adele Lu; Schoon, Boudewijn; Pouwelse, Johan A.; Epema, Dick H.J.

DOI

[10.1002/cpe.3575](https://doi.org/10.1002/cpe.3575)

Publication date

2015

Document Version

Accepted author manuscript

Published in

Concurrency and Computation: Practice & Experience

Citation (APA)

Jia, A. L., Schoon, B., Pouwelse, J. A., & Epema, D. H. J. (2015). Estimating user interaction strength in distributed online networks. *Concurrency and Computation: Practice & Experience*, 27(17), 5311-5331. <https://doi.org/10.1002/cpe.3575>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Estimating user interaction strength in online networks

Adele L. Jia, Boudewijn Schoon, Johan A. Pouwelse, and Dick H.J. Epema
Parallel and Distributed Systems Group
Department of Computer Science, Delft University of Technology, the Netherlands
Email: adele.lu.jia@gmail.com

Abstract—Online networks like Facebook and BitTorrent are based on user interactions such as wall posts and content exchange. In such systems, user relationships can be used to enhance security and promote cooperation, but in order to be meaningful, these relationships should be based on user interaction strength instead of “binary” friendships. To date, several centralized schemes for *estimating user interaction strength* have been proposed. In contrast, we present the design, deployment, and analysis of the UISE scheme for User Interaction Strength Estimation for both centralized and decentralized online networks.

Among the strong points of UISE is that it captures both direct and indirect user interactions, that it scales with only partial information dissemination in decentralized systems, and that it can be easily incorporated into distributed systems. We apply UISE to detect user interaction patterns based on wall posts in Facebook and we derive patterns that resemble those often observed in the offline human society. We further apply UISE to devise the first distributed scheme for online time estimation and we implement it into Tribler, a distributed online network for media and social applications like file sharing, streaming, and voting. We demonstrate the accuracy and the scalability of UISE with different information dissemination protocols and user behaviors using simulations, emulations, and a real-world deployment.

I. INTRODUCTION

Online networks are complex distributed computer systems that involve potentially large numbers of humans with their inputs and decisions. Typical examples of online networks include email, Facebook, LinkedIn, Wikipedia, eBay, and BitTorrent-like Peer-to-Peer (P2P) systems. They have become popular and powerful infrastructures for communication and they provide various mechanisms for users to interact. For instance, in Facebook, users post messages on their friends’ walls and comment on their friends’ photos; in Wikipedia, users collectively edit articles in their areas of expertise; and in BitTorrent, users upload to and download from each other to share the contents of their common interests. In this paper, we devise a framework for expressing user interactions and their strengths that is both generic and can be applied to a wide range of systems and applications.

The patterns and strengths of user interactions are prominent in online networks. In BitTorrent, user interactions can be used to design contribution incentive policies. Through estimating user interaction strength in terms of the amounts or durations of uploads, system designers can make users favor the highly active users for future uploads. As another example, distributed systems often rely on importing trust relationships

from social networks like Facebook to enhance security, to promote cooperation, and to improve item recommendation [14], [15], [19], [20]. Instead of leveraging static “binary” friendships, users would be much better off by estimating their interaction strengths with others and by trusting the ones with whom they have interacted frequently. Finally, sociologists often rely on user interactions for identifying social ties [8], [18], and therefore a proper estimation of user interaction strength is essential.

The importance of user interactions in online networks leads to the question: *How can we estimate user interaction strength?* Previous work addressing this issue [2], [16], [17], [18] is insufficient. First, they have focused only on online social networks like Facebook. Secondly, they have only considered *binary* and *direct* user interactions, simply indicating whether a user has directly interacted with another user or not. Thirdly, they are based on centralized algorithms that are not scalable given the astounding growth of online networks—by now, Facebook has exceeded a billion users and BitTorrent is serving hundreds of millions of users. To remedy these, in this paper we propose a distributed User Interaction Strength Estimation (UISE) scheme that has a more fine-grained notion of user interaction and that is applicable to a more general category of online networks. Specifically, we make the following contributions.

As a model for representing user interaction histories, we introduce the *bitmap-based user interaction graph*, based on which UISE estimates user interaction strengths. UISE captures the frequencies of both direct and indirect interactions and can be easily incorporated into distributed systems (Section III).

We apply UISE to detect user interaction patterns in online networks. We take Facebook as an example and we derive patterns resembling those often observed in the offline human society (Section IV). We further apply UISE to derive a scheme for estimating the time users are online in fully distributed systems. We have implemented this application into Tribler [12] and we demonstrate the scalability and the accuracy of UISE through simulations, emulations, and Internet deployment (Sections V-VIII).

As it turns out, in order to maintain the accuracy of UISE, its requirement for the coverage of global information decreases with the population size, thus allowing UISE to achieve good scalability in a self-organized manner. Further, although a user only possesses a partial view of the system, with UISE he can derive a ranking of users according to his

estimations of their interaction strength that highly resembles the ranking derived from the global view. Thus, UISE achieves the most important goal of estimating user interaction strength, i.e., differentiating users with different levels of activity.

II. USER INTERACTIONS AND DESIGN CONSIDERATIONS

User interaction strength is reflected by two aspects: the frequency of interactions and the intensity of each interaction. In this paper, we do not consider the latter aspect in order to avoid evaluating the strengths of words, such as to decide which comment should get a higher weight, “Happy birthday” or “You look nice”. Rather, in this paper we define user interaction strength as the *frequency that two users interact*, and we propose a model for estimating user interaction strength based on this definition. In this context, the following three issues are addressed.

Partial history versus full history of user interactions.

A properly selected partial history of user interactions is more suitable for estimating user interaction strength than a full history, because user interactions in online networks change rapidly over time. In Facebook, only 30% users consistently interact from one month to the next [16], and in BitTorrent, users who directly download from each other for one file rarely meet again—the so-called *problem of low rendezvous*. With this rapid change, stale interactions are no longer meaningful for inferring user relationships.

Direct versus indirect interactions. Indirect interactions are formed when users are linked through a sequence of interactions, such as in Facebook when a user posts on another user’s wall who in turn posts on a third user’s wall and so on, and in BitTorrent when a user uploads to another user who further uploads to a third user and so forth. Indirect interactions should also be included in the estimation of user interaction strength, for two reasons. First, when direct interactions are relatively scarce, such as in BitTorrent-like P2P systems where the problem of low rendezvous exists, indirect interactions provide supplementary information for inferring user relationships. Secondly, a group of direct and indirect interactions that happen within a short time frame may indicate offline relationships. In the Facebook example, the corresponding users could have participated in some offline event together and are sharing their experiences. Indirect interactions that happen widely apart in time, however, are of limited use.

We provide two options to control the level of indirect interactions to be included in the estimation of user interaction strength. The first is a *distance limit* in terms of the number of hops between interacting users. The second is an *eligible period*. Only indirect interactions that happened together with related direct interactions within this eligible period are included, e.g., when a user exchanges messages with another user who further exchanges messages with a third user, the first and third users are only linked when these interactions happened within, say one week. The distance limit and the length of the eligible period are tunable parameters, so that our design will be applicable to different applications. For example, researchers analyzing social ties can leverage our design to include different levels of indirect interactions in their models [8], [18].

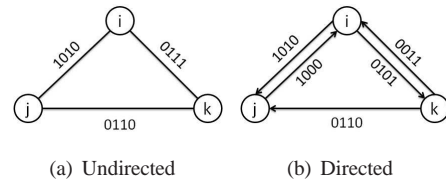


Fig. 1. Examples of user interaction graphs

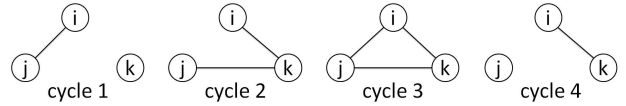


Fig. 2. Per-cycle user interaction graphs generated from Fig. 1(a).

Scalability. As an online network evolves, users involve in a huge number of interactions, which have to be collected and analyzed to estimate user interaction strengths. Doing so at central servers is neither scalable nor practical. Instead, we propose a decentralized approach in which the collection of interaction records and the estimation of interaction strengths are performed by the end users.

III. DESIGN DESCRIPTION

In this section, we introduce the basic design of UISE, a user interaction strength estimation scheme that addresses the issues listed in Section II.

A. Representing user interaction history

In UISE, users collect the whole or parts of the interaction histories of other user pairs through central servers or distributed information dissemination. The interaction histories received by a user are incorporated into its *bitmap-based user interaction graph* (UIG), a model that we introduce to represent user relationships based on their interactions. Users build individual UIGs unless they can obtain full knowledge of the system, for example, through central servers. Therefore, a UIG reflects a user’s local view of the system.

In a UIG, a vertex represents a user and the edge between two vertices represent the interaction history of the users it connects. The interaction history is reflected by the label of the edge, which is a string called the *interaction bitmap*, or simply *bitmap*. To capture the interaction frequency, we abstract time into cycles where one cycle represents a certain unit of time such as 30 minutes. We keep the interaction history in a time-based (cycle-based) sliding window fashion, with the window size being equal to the length of the kept history. When two users have directly interacted in a particular cycle, the corresponding bit in their bitmap is set to 1 (otherwise it is set to 0). As time evolves, their interaction bitmap becomes a binary string and the number of “1”s shows how frequently they have recently interacted.

User interactions can be undirected (e.g., wall posts in Facebook) or directed (e.g., uploads and downloads in BitTorrent). Fig. 1 shows examples of UIGs for undirected and directed user interactions, respectively, with a window size of 4. When Fig. 1(a) is derived from wall posts in Facebook, it specifies that users *i* and *j* have chatted on each other’s wall in

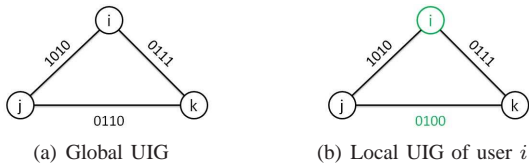


Fig. 3. An example of the global and local user interaction graphs (UIGs)

cycles 1 and 3. When the example is derived from the upload and download interactions in BitTorrent, Fig. 1(b) specifies that user i has uploaded to user k in cycles 2 and 4.

B. Estimating user interaction strength

To calculate the interaction strength, i.e., the frequency that two users interact (directly or indirectly), we perform cycle by cycle examinations. First, a UIG is divided into a number of *per-cycle* UIGs. Then, for each of these per-cycle UIGs, an algorithm for finding *connected components* or *reachability* is applied when the interactions are undirected or directed, respectively: if two users are connected (or one is reachable from the other) in a per-cycle UIG, they are considered as having interacted in that cycle. Finally, the ratio between the total number of these recognized cycles and the window size gives their interaction strength. As an example, Fig. 2 shows the per-cycle UIGs derived from Fig. 1(a). Here, an algorithm for finding connected components is applied: users i and j have interacted directly in cycles 1 and 3, and indirectly in cycle 2. Therefore, their user interaction strength is estimated to be $3/4 = 0.75$.

The requirement for being in the same connected component or being reachable serves two purposes. First, it specifies that only indirect interactions that happened together with related direct interactions within the same cycle are included in the estimation. Secondly, when UISE is applied in a distributed system, it alleviates the potential manipulations of malicious users, since users only trust the bitmaps that can link back to themselves in their UIGs. During the calculation of user interaction strength, we also provide the option of a distance limit in terms of the number of hops between interacting users. This limit is a tunable parameter for specifying the range of indirect interactions to be included in the calculation. With a limit of one hop, only direct interactions are included.

The connected components in an undirected graph and the reachability in a directed graph can be computed in linear time (in terms of the numbers of the vertices and edges of the graph) using either breadth-first search or depth-first search [10]. Thus, UISE achieves linear time complexity for estimating user interaction strength and it captures the frequency of both direct and indirect interactions.

C. Incorporating into distributed systems

We now show how UISE can be incorporated into a distributed system without central servers. We assume that in a distributed system, users can obtain information through the dissemination protocol provided by the system. In UISE, after two users have directly interacted for the first time in a particular cycle, they generate an *interaction record*, and disseminate this record into the system. Based on the

interaction records obtained through dissemination, each user builds its own *local* UIG. Further dividing a local UIG based on cycles gives the *local per-cycle* UIGs. A local UIG is a subset of the *global* UIG, in terms of the vertices, the edges, and the interaction bitmaps. The global UIG can only be obtained when the underlying dissemination protocol achieves a 100% coverage. In this paper, the 100% coverage case is used as the baseline for performance evaluation.

Fig. 3 shows an example of the global and local UIGs. Notice that user i did not receive the interaction record between j and k for cycle 3. Therefore, in its local UIG (Fig. 3(b)), the interaction bitmap between j and k is 0100, instead of the ground truth 0110.

Until now, we have introduced the basic design of UISE. In the following sections, we demonstrate and evaluate two examples of its application. In Section IV, we apply it to detect user interaction patterns in Facebook, a centralized online social network. In Sections V-VIII, we apply it to derive a decentralized scheme for estimating the online times of users and we evaluate its performance through simulations, emulations, and Internet deployment.

IV. INTERACTION PATTERN DETECTION

In this section, we apply UISE to detect user interaction patterns in online networks.

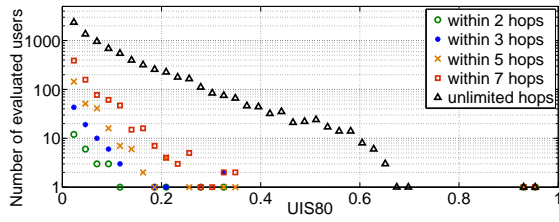
A. Applying to interaction pattern detection

We take *wall posts* in Facebook as the example of user interactions. In Facebook, each user can post messages on the walls of his friends. [16] published the entire wall post histories of 60,290 users in the New Orleans network until Jan 22, 2009. We select their data of the last year—Jan 23, 2008 to Jan 22, 2009—as the user behavior imported into our experiment. In total, 44,397 users and 876,993 posts are included.

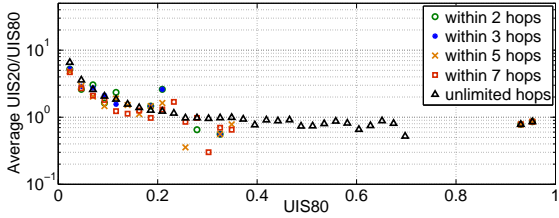
We set the cycle size to one week and we divide the data into two parts: the first 80% as training data (weeks 1-43) and the latter 20% as testing data (weeks 44-54). When a message is posted on a wall in some week, the two users involved are considered as having directly interacted, and the corresponding bit in their interaction bitmap is set from 0 to 1. As Facebook is centralized, we can obtain the interaction bitmaps of all user pairs. For each user (called the *evaluating user*), we calculate its user interaction strength (UIS) with all other users (called the *evaluated users*) based on the two parts of the data, and we refer to the results as UIS80 and UIS20, respectively. We group the evaluated users based on the value of their UIS80. We use UIS80 to demonstrate the user interaction pattern and we use the ratio between UIS20 and UIS80 (represented as UIS20/UIS80) to demonstrate the evolution of the user interaction.

B. Results

We first show the UISs of a highly active user in Fig. 4. We consider a user to be active in a particular week if he has exchanged at least one message with another user. In total, the highly active user has been active in 53 out of 54 weeks and has exchanged 2,083 messages with 25 of his friends.



(a) The number of evaluated users in each UIS80 group



(b) Average UIS20/UIS80 in each UIS80 group

Fig. 4. Interaction pattern of a highly active Facebook user (the vertical axes are in log-scale)

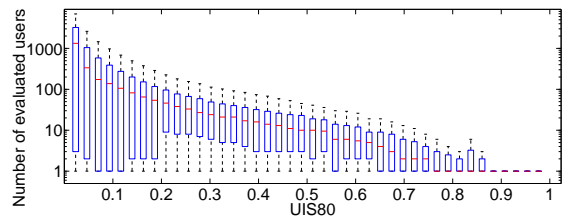
For the first step, we do not consider the distance limit, i.e., the limit of the number of hops between interacting users. Fig. 4(a) (represented by triangles) shows that this highly active user interacts intensively with only a small group: the number of users in the UIS80 groups decreases dramatically with increasing values of UIS80. A similar phenomenon is often observed in human society where people tend to interact frequently with relatively small groups and occasionally with the people outside those groups [9]. The small group could be friends, with whom people interact directly, or friends of friends, with whom people build bonds through, for example, sharing gossips with friends. On the other hands, Fig. 4(b) shows that the average value of UIS20/UIS80 decreases with increasing values of UIS80 until UIS80 equals 0.25, and stays stable (at a little bit less than 1) afterwards. This indicates that the interactions between the evaluating user and the users with whom it has high (low) interaction strengths in the first 80% of the year tend to stay stable (become more intense) in the latter 20% of the year. It can be conjectured that the same dynamic holds in offline social relationships.

To evaluate the influence of indirect interactions, we have also performed tests where we consider different distance limits. For example, for the result of “within 2 hops” as shown in Fig. 4, we have only considered the evaluating user’s friends and friends of friends. We show the results for different numbers of hops and we observe a similar tendency as for our original design without a distance limit.

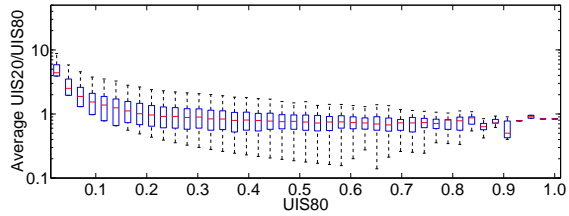
We have tested all users as in the above example and the results are shown in Fig. 5 (we do not consider a distance limit here). We find similar user interaction patterns in these results as the highly active user.

V. DISTRIBUTED ONLINE TIME ESTIMATION

Online time is essential for measuring user activity in both centralized and distributed online networks. It reflects user’s stickiness to social networks like Facebook, and it implies a user’s contribution level in P2P systems like BitTorrent. Nevertheless, online time is often recorded by central servers.



(a) Statistics of the number of evaluated users in each UIS80 group



(b) Statistics of the average UIS20/UIS80 in each UIS80 group

Fig. 5. Interaction pattern of all Facebook users with the minimum, the maximum, the median, and the 25th and 75th percentiles (the vertical axes are in log-scale)

In this section, we apply UISE to devise the first distributed scheme for online time estimation, and we introduce how to implement this application into Tribler.

A. Applying to distributed online time estimation

In this application we take *rendezvous* as the example of user interactions. *Rendezvous* means that two users meet online through, for example, a Skype video call, a BitTorrent inquiry of available files, or simply a keep-alive *ping*. It is the most fundamental user interaction in online networks. Applying UISE, when two users meet in a particular cycle, they generate an interaction record and disseminate it into the system. Based on the records received through dissemination, a user i builds its local per-cycle UIGs. If in any of these UIGs a user j is in the same connected component as i , i recognizes j to be online in that cycle. The number of these recognized cycles gives j ’s online time as estimated by i , and the user interaction strength between i and j computed as the number of these cycles divided by the length of the interaction history, gives j ’s fraction of online time as estimated by i .

As discussed in Section III-B, the requirement for being in the same connected component is for maintaining the security against malicious users, which generates a side effect of UISE—the accuracy of an evaluating user’s estimations is limited by its own online time: as we will see, the more active a user is, the more accurate his estimations will be.

B. Implementing into Tribler

We have implemented the distributed online time estimation application with UISE into Tribler [12], a fully distributed open-source online network for media and social applications like file sharing, streaming, content searching, voting, and interest-based channels. Users in Tribler interact in various ways including *rendezvous*, upload, download, and additionally, comment, vote, and spam report in channels they join. Fig. 6 shows the general architecture of the Tribler system. Tribler uses the BitTorrent protocol for P2P file sharing and

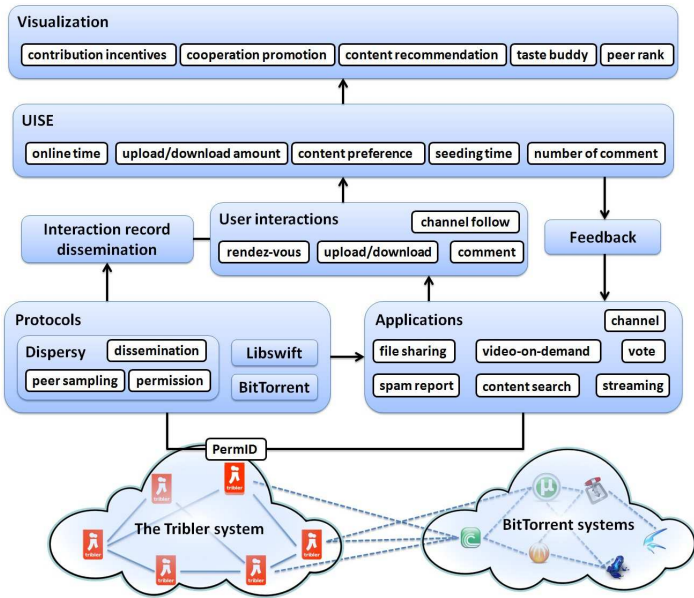


Fig. 6. The Tribler system

the Libswift protocol for P2P streaming. Libswift [11] is an IETF (Internet Engineering Task Force) standard protocol proposed by the Tribler group. We have implemented UISE into Tribler, where the underlying peer sampling and information dissemination is supported by Dispersy [21]. The estimations can be fed back to applications for policy design, such as in file sharing to reciprocate active users with priority for future downloads; they can also be visualized in the user interface to psychologically motivate users to contribute.

VI. SIMULATION

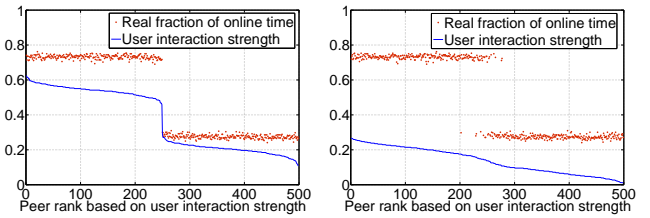
In order to evaluate the performance of UISE, we take distributed online time estimation as the example. The advantage of online time is that it is a metric with a ground truth—by comparing the real and the estimated online times, we can assess the accuracy of UISE. We address three questions: how well does UISE perform for different information dissemination protocols; how well does it perform for different user behaviors; and how well does it perform in the real world.

To answer the first question, in this section we run simulations with generic information dissemination, which allow us to explore the accuracy and scalability of UISE under different dissemination protocols by tuning the coverage. To answer the second question, in Section VII we run emulations of UISE under various real-world user behaviors. To answer the last question, in Section VIII we report measurement results derived from the Internet-deployed Tribler system.

A. Basic simulation model

Synthetic user behavior: At any time, a peer¹ can be either online or offline. When an online session ends, it starts an offline session immediately, and vice versa. The online and offline session lengths follow exponential distributions, as

¹From here, we use *user* and *peer* alternatively to refer to the functioning agent in our experiments.



(a) An evaluating peer from class A (b) An evaluating peer from class B

Fig. 7. Comparison between the real and estimated fractions of online time

observed in many distributed systems [4]. We do not consider population turnover in synthetic user behavior. Instead, later in Sections VII and VIII we use user behavior generated from measurements where population turnover is naturally included. Let (S_{on}, S_{off}) represent the average online and offline session lengths. We consider two classes of peers: (i) active peers, class A, with (8, 2) cycles, and (ii) less active peers, class B, with (2, 8) cycles.

Synthetic peer discovery and record dissemination: We abstract peer discovery and record dissemination into constant probabilities, P_1 and P_2 , respectively, and we apply them in every cycle to specify the probability that any two online peers meet and generate an interaction record, and the probability that an interaction record is received by a third peer. In the real world these probabilities should be dynamic, nevertheless, by assuming a constant probability and setting different values to it, we can analyze the performance of UISE for different peer discovery and dissemination protocols.

Simulation setup: We run each simulation for 336 cycles, i.e., 168 hours (7 days) when one cycle represents 30 minutes in the real implementation². Unless otherwise stated, we consider 250 peers in class A and 250 in class B, and we set P_1 to 20% and P_2 to 50%. Based on the synthetic user behavior and record dissemination, each peer gradually collects interaction records and builds its local UIG. At the end of the simulation, it estimates its user interaction strength with every other peer, which, as specified in Section V-A, is equal to its estimation of the fraction of online time of another peer. By comparing this estimation with the real fraction of online time, we evaluate the accuracy and scalability of UISE.

B. Accuracy

We first show in Fig. 7 the comparison between the real and estimated fractions of online times, where the latter is represented by the user interaction strengths (UISs) between the evaluating and evaluated peers. In Figs. 7(a) and 7(b), the evaluated peers are ranked according to their UISs with an evaluating peer in class A and in class B, respectively. We see that the peer in class A achieves more accurate estimations than the peer in class B. Peers in class B only stay online for $S_{on}^B / (S_{on}^B + S_{off}^B) = 20\%$ of the time and therefore they meet few peers to build their local views. As stated in Section V-A, this is a compromise for maintaining the security.

We use the Spearman Ranking Correlation Coefficient (SRCC) [13] to assess the accuracy of the peer ranks estimated

²We will give the reason later in Section VII-B

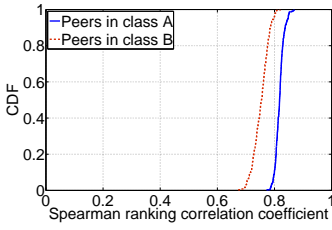


Fig. 8. CDF of SRCC in a simulation with 500 peers.

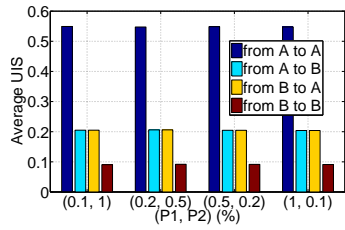


Fig. 9. Average user interaction strengths (UIS) for different values of P_1 and P_2 .

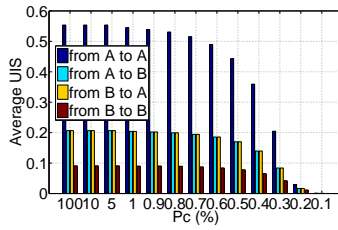


Fig. 10. Average user interaction strengths (UIS) for different values of P_C .

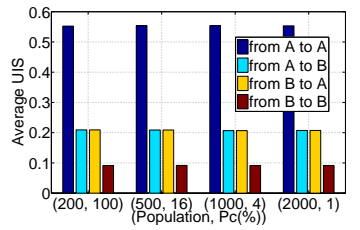


Fig. 11. Average user interaction strengths (UIS) for different populations and values of P_C .

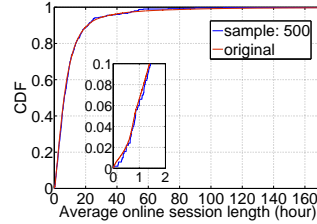
by each peer. In brief, SRCC measure the monotonic dependence between two variables. For each evaluating peer, first, we rank its evaluated peers based on its online-time estimations for them; then, in the order of this ranking, we generate two variables—a list of the real online times and a list of the online-time estimations; finally, we calculate the SRCC of these two variables. In this way, we can assess the correlation between the local rank of peers at the evaluating peer and the global rank of peers based on their real online times. For the two examples shown in Figs. 7(a) and 7(b), the SRCCs are 0.7968 and 0.7812, respectively. Fig. 8 shows the CDF of the SRCC achieved by peers in class A and in class B. We see that their local ranks of peers resemble the global one. One important application of online time estimation is to differentiate users with different levels of activity. Then, only the rank of users is needed and a design that achieves an accurate rank, like UISE, will be suitable.

C. Accuracy under partial information

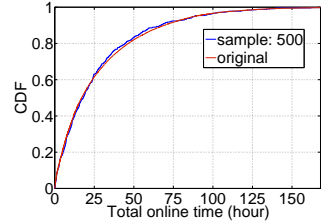
To test the accuracy of UISE under partial information, we first vary P_1 and P_2 in such a way that $P_C = P_1 \times P_2$ is constant (equal to 10%). P_C represents the probability of establishing an edge between two peers in a local per-cycle UIG of a third peer. Intuitively, it decides the third peer's estimations for others. In Fig. 9 we show the user interaction strengths (estimated fractions of online time) averaged over the classes of evaluating (“from” in the figure) and evaluated (“to” in the figure) peers. We find that, consistent with our intuition, the estimations stay stable for different values of P_1 and of P_2 while P_C is constant. This allows us to analyze the influence of P_C without exploring extensive combinations of P_1 and P_2 —We keep P_1 constant (equal to 1) and vary P_2 in such a way that P_C is decreased from 100% to 0.1%. In Fig. 10 we show the user interaction strengths averaged over classes. We find that when P_C is at least equal to 5%, i.e., when peers hold at least 5% of the total information, the estimations stay stable. Further reducing P_C results in noticeable decreases of estimations, nevertheless, UISE still achieves a decent user rank: on average, peers in class A are correctly estimated as more active than peers in class B.

D. Scalability

In this section we test the scalability of UISE under different populations. Let $N(t)$ represent the number of online peers at cycle t . At the end of the simulation, for cycle t , each peer will receive $N(t)(N(t) - 1)P_C$ records and will generate $N(t)(N(t) - 1)P_C$ edges in its local per-cycle UIG. To test



(a) CDF of session length



(b) CDF of total online time

Fig. 12. Comparison between the original FileList trace and our sample.

the scalability, when the population increases, we decrease P_C in such a way that $N(t)(N(t) - 1)P_C$ is constant.

In UISE, an evaluating peer recognizes another peer to be online at cycle t if they are in the same connected component in its local per-cycle UIG for cycle t . In graph theory, for a random graph with n vertices to be connected, the expected number of edges needed is less than $n \ln n$ [10]. Therefore, the basic condition for a peer to correctly recognize all the peers online at cycle t is:

$$N(t)(N(t) - 1)P_C \geq N(t) \ln(N(t)) \Rightarrow P_C \geq \frac{\ln(N(t))}{N(t) - 1}.$$

As $\ln(N(t))$ increases very slowly with $N(t)$, the required value for P_C decreases strongly with the population. Thus, UISE achieves good scalability in a self-organized manner.

The simulation result confirms the above analysis. Fig. 11 shows that the average user interaction strengths (estimated fractions of online time) stay stable while we increase the population and decrease P_C accordingly.

VII. EMULATION

In this section we evaluate the performance of UISE for different user behaviors. As we have tested different dissemination protocols in Section VI, we now release the assumption of a generic information dissemination and we test UISE under Tribler's dissemination protocol based on emulations. Due to the limit of space, we only show results with user behaviors generated from measurement traces. More results about different churn patterns of users can be found in our technical report [7].

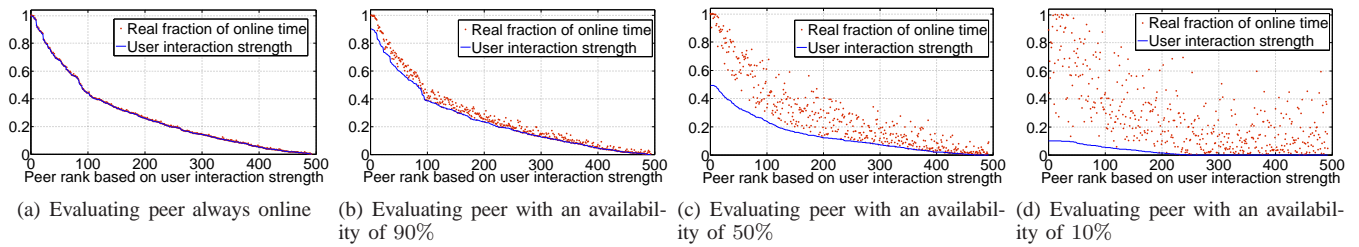


Fig. 13. Comparison between real and estimated fractions of online time for FileList trace

A. Emulation setup

The emulation is performed on an anonymous cluster that contains 23 nodes, each of which has two 2.4 Ghz quad-core processors and 24 GBytes of memory. The nodes are connected by a 10 Gb/s QDR Infiniband interconnect. We import the online and offline behaviors of users from measurement traces generated from the FileList BitTorrent community [1]. These traces contain uptime and downtime of every user that was online at least once during the measurement period. In total, we captured 63,548 users in 7 days, from which we randomly select 500 users for our emulations. Fig. 12 shows the CDFs of the average online session length and the total online time for the original trace and our sample. We see that our sample represents the original trace well. Further, as 94% of the online sessions are longer than one hour, we set the cycle size to 30 minutes in the real implementation, so as to capture most of the online sessions.

In our emulation, peers run Tribler’s dissemination protocol: they meet, generate and disseminate interaction records, and store the records received from dissemination in their local SQLite databases. At the end of each emulation, they estimate their user interaction strengths with others, which give their estimations of the fractions of online time of others. We set the cycle size to 2 minutes and we run each emulation for 10 hours, resulting in interaction bitmaps of $10 \times 60/2 = 300$ bits. The small cycle size and short emulation time are compromises for the time consumption of the cluster. This parameter setting represents a running time of 7 days when the cycle size is set to 30 minutes in the real world implementation.

B. Results

Fig. 13 shows four examples of comparisons between the real and estimated fractions of online time, where the latter one is represented by the user interaction strengths between the evaluated and evaluating peers. In Figs. 13(a), 13(b), 13(c), and 13(d), the evaluated peers are ranked based on their online times estimated by a peer with an availability of 100%, 90%, 50%, and 10%, respectively. The first three peers achieve estimations very close to the real fractions of online time, with an SRCC equal to 0.9998, 0.9945, and 0.9388, respectively. We can see a clear decrease of the real fractions of online time when the evaluated peers are ranked based on the estimations from these three evaluating users, indicating that their local ranks of peers closely resemble the global one.

Evaluating peer 4 (Fig. 13(d)), however, only achieves an SRCC equal to 0.6853. The reason is that, as stated in Section V-A, in UISE an evaluating peer only trusts the interaction records that can link back to itself (reflected by

being in the same connected component in a local UIG). Therefore, its estimation for another user in fact reflects their *concurrent* online time. As evaluating peer 4 is only online for 10% of the time, it achieves a low accuracy. Nevertheless, its estimation for another user can be used to assess their availability to each other—an important issue for distributed online networks where users collaborate and only the ones online simultaneously can help each other. In Fig. 14 we show evaluating peer 4’s estimations and its fraction of concurrent online time with other peers, where we observe very accurate estimations with an SRCC equal to 0.9973.

C. Preparing for the real world

As we found in Section VI-C, UISE achieves good estimations even with limited information. This desirable feature allows us to introduce a practical strategy, *targeted-generation*, to reduce the number of interaction records to be disseminated. In targeted-generation, for each cycle, a user only generates records with the N_G users that it has observed to be online the longest during the past M cycles, resulting a constant number of records being generated per user per cycle. We run emulations to test its performance, where we use the same synthetic user behaviors as in Section VI-A. Fig. 15 shows the user interaction strengths (estimated fractions of online time) averaged over classes for different values of N_G (we set $M = 1$ in our emulations). We see that UISE performs stably when N decreases from 10 to 1, indicating that we can decrease the workload dramatically without deteriorating the accuracy of the estimation.

VIII. REAL-WORD DEPLOYMENT

We have implemented UISE into Tribler. Due to the limit of space, in this section, we only report measurements of the evolution of user’s online time and the accuracy of online time estimation. More results about the overlay structure and the performance of interaction record generation can be found in our technical report [7].

A. Deployment techniques

In the real world deployment, the cycle size is set to 30 minutes and the interaction history is kept in a sliding window fashion with a window size of 7 days. We adopt the *targeted-generation* version of UISE as introduced in Section VII-C, with $N_G = 5$ and $M = 1$. In addition, we specify that two users generate their first interaction record only until they have seen each other online for at least two cycles. This effectively prevents “hit-and-run” users generating records that are of limited use.

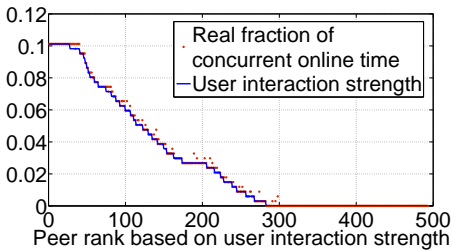


Fig. 14. Comparison between fraction of concurrent online time and user interaction strength.

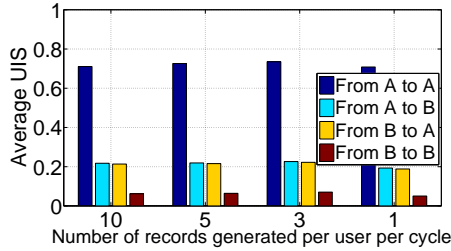


Fig. 15. User interaction strength (UIS) under targeted-generation for different values of N_G .

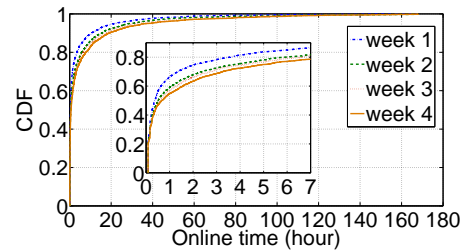


Fig. 16. CDF of online time of Tribler users.

Tribler is fully distributed, containing no central servers and hence no records of user behaviors from the global view. To obtain the ground truth for our experiment, we deploy log servers and every 5 minutes, each user reports its online activity to one of them, including its identifier, its timestamp, the number of interaction records it generated successfully, and the updated information about interaction records of other user pairs it received since last report. In total, for the first week of Tribler’s new release, we obtain 2,874 active users with unique identifiers, among which 1,713 users have generated at least one interaction record. For weeks 2, 3, and 4, we obtain 2,673, 2,905, and 2,884 active users with unique identifiers, respectively.

B. Evaluation

Evolution of online time: The dashed blue line in Fig. 16 shows the CDF of online times of Tribler users obtained from log servers during the first week. Around 15% users are online for more than 7 hours, resulting in an average of more than one hour per day. Nevertheless, 60% users are online for less than one hour in total. Comparing Figs. 16 and 12(b), clearly users in FileList are more active than users in Tribler. FileList constantly monitors user activities through central servers and it specifies that users with high contribution levels will be rewarded with the preference for future downloads, and therefore users are incentivized to stay online longer—UISE serves exactly the same goal, and moreover, it is performed in a distributed manner. As a matter of fact, in the current release of Tribler, users are educated that their activities will be evaluated through UISE. Though for now the estimated online times are not utilized explicitly as in FileList, we can already observe a gradual increase of user’s online time from week 1 to week 4 (Fig. 16). This promising observation indicates that, being aware of their activities being evaluated, users in Tribler are becoming more committed—a behavioral change that has been observed by sociologist and psychologists under similar circumstances in human society [5].

The accuracy: Fig. 17 shows the comparisons between the real and estimated fractions of online time of each Tribler user, where the latter is represented by the user interaction strengths between the evaluating and the evaluated users. In Figs. 17(a), 17(b), and 17(c), the evaluated users are ranked based on their user interaction strengths with a user with an availability of 90%, 50%, and 10%, respectively. Similar to the results of Filelist trace relay in Section VII-B, the accuracy of the estimation is limited by the availability of the evaluating user: the more active it is, the more accurate its estimations will be. In total, the three evaluating users have successfully

identified 976, 745, and 217 users to be online for at least one cycle; and the SRCCs between their estimations and real online times are equal to 0.9325, 0.8635, and 0.6446, respectively. Though evaluating user 3 (Fig. 17(c)) achieves a low accuracy, as shown in Fig. 18, it achieves accurate estimations of its concurrent online time with other peers (with an SRCC equal to 0.9785). Thus, it can successfully identify the users with whom it is online simultaneously, i.e., the users with whom it has collaborated and potentially will collaborate.

IX. RELATED WORK

To date, a few works have focused on understanding user interactions in online social networks. Moon *et al.* [2] investigate the guestbook logs of Cyworld and they show that interactions between friends are highly reciprocated. Viswanath *et al.* [16] study the evolution of user interactions in Facebook and they find that user interactions change rapidly over time. These observations provide the foundation for designing UISE that only considers recent user interactions. Wilson *et al.* [17] introduce an interaction graph. They show that interaction links exhibit different properties than social links (friendships) and are more representative for inferring user relationships. Nevertheless, their interaction graph is unweighted and does not take the interaction frequency into account as we do.

Another direction of related research is identifying social ties. Kahanda *et al.* [8] propose an approach for identifying the weak and the strong ties. They focus on supervised learning models that require human annotation of link strength such as top friend nomination. Xiang *et al.* [18] develop an unsupervised model that represents a range of tie strengths based on user interactions and profile similarity. However, they consider only direct and binary interactions. Instead, we propose UISE that captures the frequency of both direct and indirect interactions, which can be leveraged by these models for identifying social ties. Moreover, while all the above related works are centralized, UISE is applicable in distributed systems.

There are also studies on leveraging user interactions in distributed online networks for policy design. BitTorrent [3] clients constantly monitor their direct interactions (uploads) with others and reciprocate the ones from whom they download the fastest. However, in BitTorrent systems the problem of low rendezvous exits and direct interactions are insufficient for inferring user relationships [6]. Meulpolder *et al.* introduce BarterCast, a distributed reputation system that ranks users based on their upload and download activity in P2P file sharing. BarterCast captures both direct and indirect user

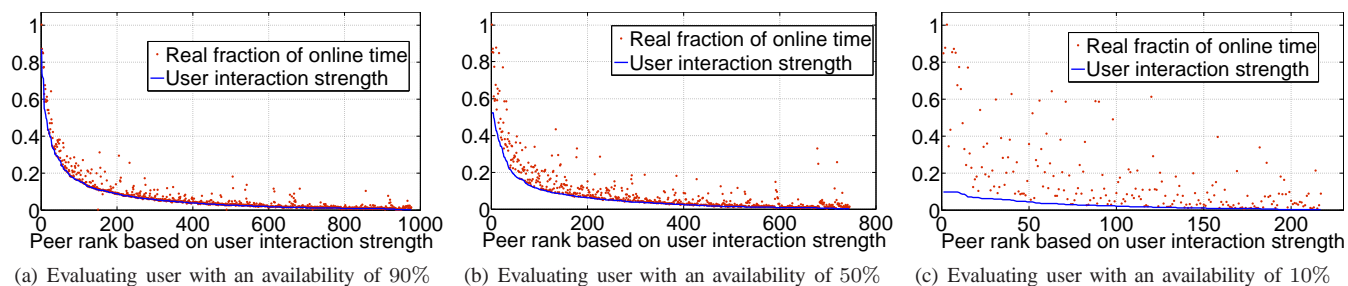


Fig. 17. Comparison between real and estimated fractions of online time

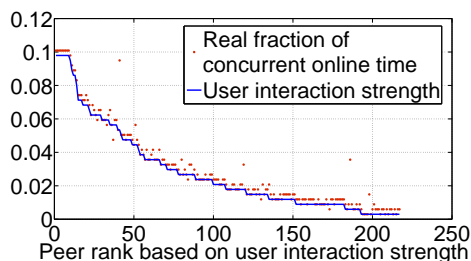


Fig. 18. Fraction of concurrent online time versus user interaction strength

interactions, however, it adopts a MaxFlow-based algorithm with a heavy complexity. Instead, UISE adopts an connect-component-based algorithm and achieves a linear time complexity in terms of the number of user pairs. We have also applied UISE to derive a distributed scheme for online time estimation. To the best of our knowledge, this is the first work that sheds lights on this topic.

X. CONCLUSION

User interaction is the most important underpinning of online networks, in which hundreds of millions of users communicate, interact, and share their online lives. In this paper we propose UISE, a scalable scheme for estimating user interaction strength in both centralized and distributed online networks. We have applied UISE to detect user interaction patterns in Facebook based on wall posts, and we have derived patterns that resemble the ones often observed in the offline human society. We have further applied UISE to design and deploy a decentralized scheme for online time estimation based on rendezvous. In the latter application we shown that UISE is scalable and stable for different dissemination protocols and for different user behaviors. We have incorporated this application into Tribler, and we have shown through Internet deployment that UISE effectively differentiates users with different levels of activity, and thus, accomplishes the most important goal of estimating user interaction strength.

As future work, we plan to apply UISE to more types of interactions than only wall posts to detect interaction patterns and rendezvous to estimate online times. For instance, we intend to explore chats over photos to friendship recommendation and upload and download to incentive policy design.

REFERENCES

[1] P2P Trace Archive. <http://p2pta.ewi.tudelft.nl/pmwiki/>.

[2] H. Chun, H. Kwak, Y.H. Eom, Y.Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in colum vs interaction: a case study of cyworld. In *Proceeding of IMC'08*, 2008.

[3] B. Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, USA, 2003.

[4] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *Proceedings of IMC'05*, 2005.

[5] B.A. Jacob. Accountability, incentives and behavior: The impact of high-stakes testing in the chicago public schools. *Journal of public Economics*, 89-5:761–796, 2005.

[6] A.L. Jia, R. Rahman, T. Vinko, J.A. Pouwelse, and D.H.J. Epema. Fast download but eternal seeding: the reward and punishment of sharing ratio enforcement. In *Proceeding of IEEE P2P'11*.

[7] A.L. Jia, B. Schoon, J.A. Pouwelse, and D.H.J. Epema. Estimating user interaction strength in online networks. Technical Report PDS-2013-007, Delft University of Technology, 2013.

[8] I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *Proceeding of ICWSM'09*.

[9] P.V. Marsden. Core discussion networks of Americans. *American Sociological Review*, 52(1):122–131, 1987.

[10] R. Motwani and P. Raghavan. Randomized algorithms. *Cambridge University Press*, 1995.

[11] R. Petrocco, J. Pouwelse, and D.H.J. Epema. Performance analysis of the libswift p2p streaming protocol. In *Proceeding of P2P'12*, 2012.

[12] J.A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M. van Steen, and H.J. Sips. Tribler: A social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 20:127–138, 2008.

[13] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72-101, 1904.

[14] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control.g. In *Proceeding of INFOCOM'11*, 2011.

[15] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *Proceeding of NSDI'09*, 2009.

[16] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *ACM SIGCOMM WOSN*, 2009.

[17] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceeding of Eurosys'09*.

[18] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceeding of WWW'10*, 2010.

[19] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy '08*, 2008.

[20] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceeding of SIGCOMM'06*, 2006.

[21] N. Zeilemaker, B. Schoon, and J. Pouwelse. Dispersy Bundle Synchronization. In *TU Delft PDS, Technical Report: PDS-2013-002*, 2013.