

Does Location Matter? The Efficiency of Request Propagation Based on Location in Online Social Networks

Salem Othman^(✉), Javed I. Khan, and Fatema Nafa

Networking and Media Communication Research Laboratories,
Department of Computer Science, Kent State University, Kent, OH 44240, USA
{sothman, javed, fnafa}@kent.edu

Abstract. The centrality metrics such as Closeness and Betweenness in Online Social Network (OSN) determine how much end-to-end delay and queue-load of a node can have as a source or as a destination through Social Routing. Experimentally, we find that nodes with high Out-Closeness centrality in OSN suffer from high end-to-end delay as a target, but not as a source. We show that the cause of this end-to-end delay is that most nodes with high Out-Closeness centrality have low In-Closeness centrality. Moreover, we show that the increase in the local In-Degree centrality will increase the global In-Closeness centrality. We also find that the promised level to increase the In-Closeness centrality of a node is its Friends of Friends-Of-Friends (Level-3). An agent-based Model for Social Routing is proposed and a set of large-scale Google+ Graphs are used. A simulation study is also completed by propagating a set of requests in different societies with different routing schemes and diverse queue disciplines, in order to compare the average end-to-end delays from the source and target perspectives.

Keywords: Online social networks · Requests · Social routing and forwarding · Simulation of online social networks · Network centrality

1 Introduction

Social Routing [14] is an important component of Online Social Networks (OSNs) (e.g. Facebook, Twitter, and Google plus). The main goal of the social routing protocols such as our SOR protocol [14] is to help individuals to communicate with indirect neighbors by disseminating their requests in the OSNs. However, the location of individuals in OSNs can be a double-edged sword. Thus, it is imperative to study and understand how the node position in OSNs can affect the efficiency of message propagation and how nodes can cope with the increase in their queue loads and in their requests delays as a source or as a destination. Source individuals who request donations from others hope to be near them while those who are in the target position might try to be out of the way temporarily or forever.

Identifying the location (centrally) of a node in OSNs has been introduced in social science to analyze the importance of nodes in social networks [7]. The three most popular individual centrality measures are Degree centrality, Betweenness centrality, and closeness centrality [7, 9]. Other definitions of closeness centrality are possible based on the

direction of the edge such as In-Closeness centrality and Out-Closeness centrality which determine how close a particular node is to its incoming and outgoing neighbors respectively [4]. Recently, increased attention has been given to identifying the centralities in blogs, wikis, social annotation and tagging, media sharing, transportation, Brain and Online Social networks. These central nodes represent the most critical nodes that can have a major impact on the network operations.

In this paper, we therefore propose an agent-based Model for Social Routing which aims to mimic human interaction in OSNs. A set of Google+ social network large-scale Graphs with a set of human attributes are used for experimental studies. A simulation study using our protocol SOR [14] is conducted by propagating a set of requests in four different societies in order to compare the average end-to-end delays from the source and target perspectives. Experimentally, we find that (1) nodes with high Out-Closeness centrality in OSN suffer from high end-to-end delay as a target, but not as a source, (2) the cause of this end-to-end delay is that most nodes with high Out-Closeness centrality have low In-Closeness centrality, (3) the promised level to increase the In-Closeness centrality of a node is its Friends of Friends-Of-Friends (Level-3).

The rest of the paper is organized as follow: Sect. 2 presents the related work. Section 3 introduces the notation and formally defines the Closeness and Betweenness centrality metrics. Our agent-based model of Social Routing, SPPD Metric, and four societies are explained in detail in Sect. 4. The simulation and experimental results are given in Sect. 5 and the conclusion is in Sect. 6.

2 Related Work

Node position within the OSN has been studied in many domains in order to analyze important nodes. In Brain Networks, node position help to Identify and Classify Hubs [15]. In blogs, it is meant to identify influential bloggers in a blogging community [11]. In Transportation Networks, it is to determine critical nodes in order to improve the design of the network and devise plans for coping with network failures [6]. In computer networks (Internet), node position is studied in order to protect against threats of central nodes [5] and to analyze country-level routing which helps to answer questions about the influence of each country on the flow of international traffic [10]. In Online Social Networks, node position helps to evaluate the centrality of the node within the social network [3], to identify the social hubs which are nodes at the center of the influential neighborhoods [8], to examine the relationship between the type of flow and the differential importance of nodes with respect to the speed of traffic reception and the frequency of receiving it [2], and to evaluate ways of predicting a node's future importance under degree, closeness, and Betweenness centralities [12]. Our work is to investigate the efficiency of request propagation based on the node's location (In-Closeness, Out-Closeness, and Betweenness centralities) in the OSN.

3 Background

In this section, we formally define the Online Social Network, In-Closeness centrality, Out-Closeness centrality, and Betweenness centrality.

3.1 Online Social Network (OSN) Model

As shown in Fig. 1, Let $G = (V, E)$ be an OSN modeled as a directed graph where V is a set of n nodes and E is a set of m edges in the graph. Let $e_{u,v}$ denote a link (social relationships) of the graph connecting a pair of nodes (u, v) and let $P_{u,v}$ denote a path between the source node u and the destination node v ; the path consisting of a series of intermediate nodes (u, I_0, I_1, \dots, v) . Let $L_{anc}(v)$ be the set of predecessors (ancestors) which are connected to v in G , let $L_{neb}(v)$ be the set of incoming direct neighbors connected to v , let $L_{src}(v)$ be the set of source nodes sending their requests to v , let $L_{int}(v)$ be the set of intermediate nodes between node v and its sources, and let $L_p(v)$ be the set of paths which start from the source nodes and end at v as a target.

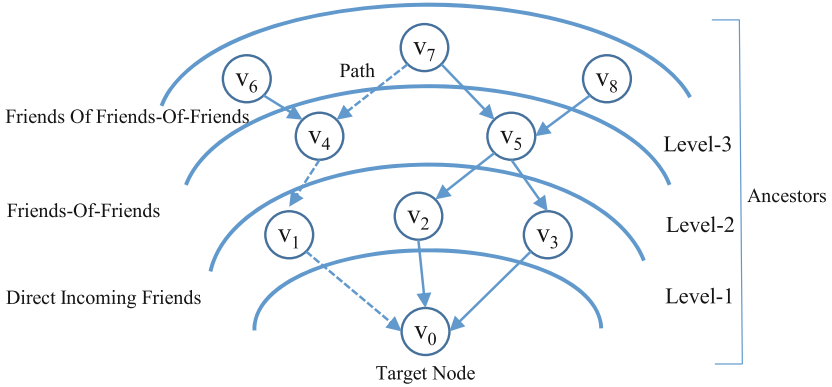


Fig. 1. Target node v and its ancestors

3.2 Node Centrality in OSN

The three most widely used centrality measures are Degree, Closeness, and Betweenness.

Closeness centrality: is a measure of the mean geodesic distance between a node and all its reachable nodes. Therefore, it identifies node location. It also refers to how near a node is to all other nodes in the network. In a social network context, this means how fast this node can reach everyone in the network; which affects the rate at which the information is propagating throughout the network [2].

The used graph is directed, so we define two versions of closeness centrality which are In-Closeness centrality (In-CC) and Out-Closeness centrality (Out-CC) measures. The difference between them is that the node with high Out-Closeness centrality is close to its outgoing neighbors and the one with high In-Closeness centrality is close to its incoming neighbors.

Betweenness centrality: measures how important a node is by counting the number of the shortest paths that pass through a node. Therefore, it measures the load of a given node. In a social network context, it means how likely a node is to be the most direct path between two individuals in the network and how it can influence the flow of information between them [2].

4 Agent-Based Model for Social Routing

For the simulation of human interaction on the OSN, we propose an agent-based model with (1) **Social-based Human-Queue model** which is a way of governing how requests are buffered while waiting to be transmitted to the next hop or to get a service. We assume that each agent in OSN has two queues Forwarding and Servicing. Furthermore, in this paper we only focus on two queuing disciplines: First-Come-First-Service (FCFS) and Social Priority (SP) as proposed by Barabasi [1]; (2) **Social-based forwarding** which is simply a social characteristic-based scheme of choosing the next hop from neighbors to receive a particular request. The next hop could be either the destination of the request or an intermediate node in a path to the destination; (3) **Social-based Routing** which is the process of exploiting the social characteristics of nodes in OSN in order to make a better routing decision by finding the best path from source to destinations; (4) The agent used **Request** which is a special kind of message containing a source, a destination, and some information which determine the request type such as LinkedIn's request for endorsements, and Facebook's request to join in a Cause, etc.

4.1 Agent Architecture

As shown in Fig. 2, each agent is associated with (1) a queue named forwarding queue and denoted as Q_u^f which is a data structure for storing requests temporarily. (2) A queue manager which utilizes queue disciplines for inserting, dropping, popping, and ordering requests. The Forwarding queue has three parameters (a) request arrival rate, λ_u^f , which is the number of requests arriving at u 's queue per unit time, (b) request forwarding rate, μ_u^f , which is the number of requests departing the u 's queue per unit time, and (c) forwarding queue length, $L_u^f(t)$, which is the number of requests in the forwarding queue of node u at time t . (3) The forwarding manager which forwards the requests to the next neighbor based on forwarding strategies.

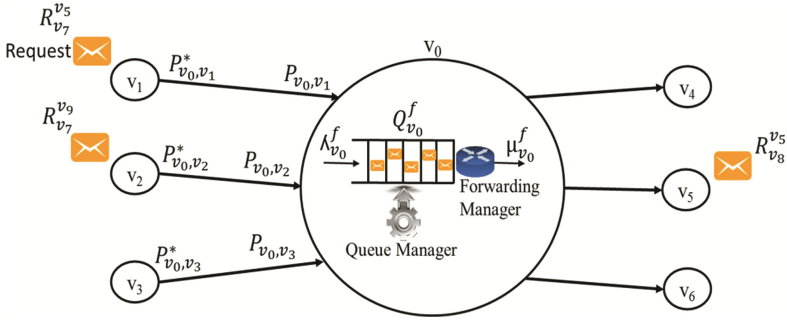


Fig. 2. Agent anatomy

4.2 Social Priority (SP)

The friendship edge, $e_{u,v}$, from node u to node v is associated with two values as shown in Fig. 2: an In-Social Priority (iSP) for forwarding and it represents a form of proportionate priority with which v will treat a request arriving from u and an Out-Social Priority (oSP) for determining the best path to forward. The value of iSP is known to v but, but we cannot expect v to reveal it candidly to u , thus, u continually learns oSP which is an estimate of iSP . If node u makes a correct estimation then $oSP = iSP$. In our model, we factor namely Gender, Degree, Betweenness, Closeness, Eigenvector centralities, etc. In order to generate social priorities for all potential senders (receivers), each node uses its own set of factors and uses singular value decomposition (SVD) [13] to generate a SP vector for the immediate in(out) circle made of adjacent neighbors.

4.3 Social Priority Based Path Delay (SPPD) Metric

In this subsection, we describe the SPPD metric and the information needed for it. The objective is to determine the end-to-end delay, $T_{\text{end-to-end}}$, experienced by a request R_s^d through simple and autonomous paths from a source node s to a destination node d . We assume (1) the node can use only one queue discipline (SP or FCFS) which is known to all nodes in the network, (2) the node u 's queue parameters (λ_u^f , μ_u^f , $L_u^f(t)$) are also known. Table 1 summarizes the used parameters.

Table 1. Queue parameters

Parameter	Description
$L_u^f(t)$	The number of requests in forwarding the queue of node u at time t .
λ_u^f	The number of requests arriving at u 's queue per unit time
μ_u^f	The number of requests departing the u 's queue per unit time
$oSP_{u,v}$	Out-social priority
$\beta_u(t)$	The queue discipline of node u at time t . It is fixed all times?

Given a request $R_{v_1}^{v_k}$, a simple path $P_{v_1, v_k} = (v_1, \dots, v_k)$, and all parameters of intermediate nodes as depicted in Table 1, **find the end-to-end delay that the expected request will experience through a given simple path.** To find it, we introduce Eqs. 1 and 2:

$$T_{v_i}^f = \frac{L_{i+1}^f(t_0) + T_{v_{i-1}}^f * \lambda_{v_{i+1}}^f}{\frac{\mu_{v_{i+1}}^f}{oSP_{v_{i+1}, v_i}} - \lambda_{v_{i+1}}^f} + \frac{oSP_{v_{i+1}, v_i}}{\omega} \quad (1)$$

Where $T_{v_0}^f = 0$ and $i = 1, 2 \dots v_{k-1}$, ω is a constant value.

Generally, the end-to-end delay for any number of intermediate nodes in the simple path is computed by Eq. 2:

$$T_{end-to-end}(v_1, v_k) = \sum_{i=1}^{k-1} T_{v_i}^f + c \quad (2)$$

The above equations can be modified to calculate FCFS queue discipline by putting $oSP_{v_{i+1}, v_i} = 1$ which means the position of the request will be at the bottom of the queue regardless of how the out social priority is.

4.4 Societies in Online Social Networks

In this section we introduce a set of societies in OSN according to routing algorithms, the queuing discipline, and the forwarding schemes of nodes in OSN. We propose four societies. Two of them use the Social Priority-based routing algorithm and the others use the First-Come-First-Service based routing one. Moreover, the Social Priority-based queue and the First-Come-First-Service based queue disciplines are used to study the misalignment between routing algorithms and the queuing discipline. Let us discuss the differences and similarities between the four societies.

SP-SP society: In this social system, the source nodes use Social Priority-based routing algorithm to get the best paths to a particular target and the intermediate nodes use Social Priority-based queue discipline. This is an ideal society where there is no misalignment between the routing algorithms and queue disciplines, and where computing the best path to any target is possible and accurate.

SP-FCFS society: In this mock community, the source nodes use Social Priority-based routing algorithm in order to get the best paths to a particular target and the intermediate nodes use First-Come-First-Service based queue discipline. This is a factual society where there is a misalignment between the routing algorithms and queue disciplines and where computing the best path to a target is possible but not as expected because of the misalignment.

FCFS-SP society: In this group, the source nodes use First-Come-First-Service based routing algorithm in order to get the shortest paths to a particular target while the

intermediate nodes use Social Priority-based queue discipline. The source nodes just know the number of hops between them and their destinations. They do not have access to critical information like the Social Priorities of nodes. However, the members of this society use Social Priority-based queue discipline which means there is a misalignment between the routing algorithms and queue disciplines.

FCFS-FCFS society: In this society, the source and intermediate nodes use First-Come First-Service based on routing and queue discipline. This is another example where the source nodes just know the number of hops between them and their destinations and hence there is no misalignment.

5 Experimental Results

In our previous work [14], we design and implement SOR using Omnet++ [16] to simulate human behavior and we also develop the Behavioral Data Analyzer (BDA) in Python and Apache Pig¹ to analyze and interpret the results. We perform a set of experiments to evaluate the efficiency of message propagation in the four societies: SP-SP, SP-FCFS, FCFS-SP, and FCFS-FCFS. We run all the experiments in amazon web services (AWS) using the instance of type t2.large². For experiments, we use real social network datasets of different sizes of Google Plus platform. Table 2 lists the datasets (graphs) along with the vertex count, the edge count, the average degree, and the graph diameter. The graphs are directed and are available in the following web sites³.

Table 2. Statistical information of Google + datasets

Dataset	#Nodes	#Edge	Avg. node degree	Graph diameter
DS-1	54	252	9.3	5
DS-2	120	1255	21.6	5
DS-3	498	13119	52.0	5
DS-4	1079	51953	96.0	6
DS-5	1648	166291	201.8	7
DS-6	2211	93509	84.5	8

We examine the impact of a node position (Closeness and Betweenness centrality) in OSN on the average end-to-end delay from the source and target perspectives. We use two performance metrics: *Target-based average end-to-end delay* ($T_AVG_EtoE_Delay$), and *Source-based average end-to-end delay* ($S_AVG_EtoE_Delay$) to compare delays in the four societies. In general, the end-to-end delay of a request in a network is the time it takes the request to reach the destination from the time it leaves the source. Each source sends a set of requests to a different target and each target receives a set of requests from different sources. We compute the Target-based average end-to-end delay and Source-based average

¹ <https://pig.apache.org/>.

² <https://aws.amazon.com/ec2/instance-types/>.

³ <http://snap.stanford.edu/data/index.html>.

end-to-end delay for each node separately by first getting all the node's received and sent requests and finding the end-to-end delays of these requests, then finding the average value of these delays. In our study delay comprises queuing and forwarding delays but not the service (processing) delay. Since we are focusing on routing and forwarding behaviors, we assign one millisecond to all service components. We use Closeness and Betweenness to measure the varying importance of the nodes in OSN.

Simulation setup: There are n forwarding queues in OSN, each with a single node. Requests generate in exponential distribution with a mean 15 min, forward in the exponential distribution with mean 5 min, and serve in 1 ms. We fix the number of requests for all of the experiments and we use the same sources and destinations. We notice that some scientists directly use distributions (e.g. Poisson or Exponential) in the simulator which may cause a different number of requests and/or different destinations. We use six datasets and for each dataset we have done four experiments (SP-SP, SP-FCFS, FCFS-SP, and FCFS-FCFS).

5.1 Source and Target Nodes Perspectives

We compare the Target-based and Source-based average end-to-end delays of SP-SP, SP-FCFS, FCFS-SP, and FCFS-FCFS societies using various datasets (graphs). Since the patterns of Target-based and Source-based average end-to-end delays of the four societies are similar in all datasets and because of the space limitation, we only show the Target-based and Source-based average end-to-end delays of the society SP-SP of the dataset DS-6 as shown in Figs. 3. The Figure shows that among the correlation between In-Closeness, Out-Closeness, and Betweenness Centralities and the Target-based and Source-based average end-to-end delays of SP-SP society, the patterns of In-Closeness centrality and Betweenness centrality are more similar to each other where nodes with high centrality get less delay than with Out-Closeness centrality. The nodes with high Out-Closeness centrality are close to their outgoing neighbors and this helps them as sources as shown in Fig. 3(c) while the same nodes with high Out-Closeness centrality suffer from very high Target-based average end-to-end delay as shown in Fig. 3(d). The following subsection introduces why nodes with high Out-Closeness centrality have this delay and how delay can be handled by nodes.

5.2 Closeness and Betweenness Centralities

We find, as shown in Fig. 4(a) where the x-axis is the position of nodes ordered by Out-CC and the y-axis is the difference between node's position in Out-CC and its position in In-CC (*Diff*), that most nodes with high Out-Closeness centrality have low In-Closeness centrality in OSN and this explains why these nodes suffer from delay as targets. We first order nodes by their Out-Closeness centrality, then we find the position of each node in the ordered In-Closeness centrality list and finally we calculate the difference between the two positions. For example, the highest Out-Closeness centrality node is node number 467 and its position is 1, however, its position in the ordered In-Closeness centrality list is 182, so, the difference is $1 - 182 = -181$. If we carefully observe Fig. 4(a)

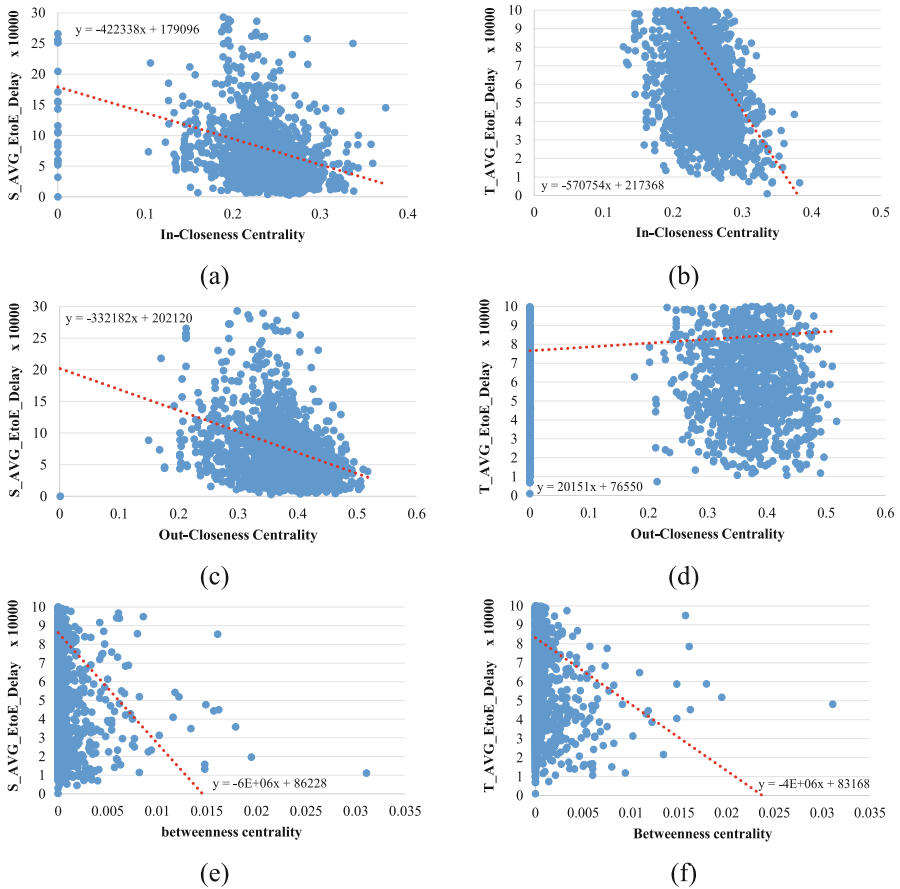


Fig. 3. Correlation between source-based average end-to-end delay and In-Closeness, Out-Closeness, and betweenness centralities in the left column and correlation between Target-based average end-to-end delay and In-Closeness, Out-Closeness, and betweenness centralities in the right column.

we can also see that some nodes with low Out-Closeness centrality have high In-Closeness centrality. Nodes can handle this problem by increasing their In-Closeness centrality and adding more edges. In Fig. 4 (b) we find the node with high local In-Degree centrality (which is a count of the number of links directed to the node) also has high global In-Closeness centrality. This pattern can be used to encourage nodes to increase their local In-Degree centrality to be globally important (High In-Closeness centrality) in the OSN which will help them to reduce the Target-based average end-to-end delay as shown in Fig. 3(b).

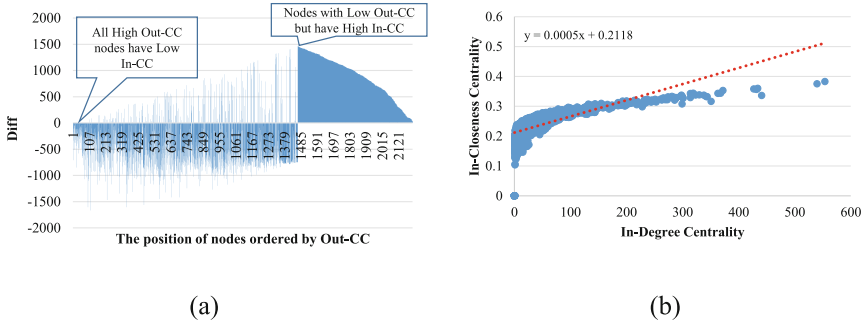


Fig. 4. (a) The difference between Out-Closeness centrality and In-Closeness centrality positions of nodes and (b) The correlation between In-Degree and In-Closeness Centralities

5.3 Level Ancestor and in-Closeness Centrality

We define a level as a distance (number of edges) from an ancestor node u to a target node v . Thus as depicted in Fig. 1, ancestors in level-1 are v 's direct incoming neighbors, ancestors in level-2 are v 's incoming friends-of-friends, ancestors in level-3 are v 's incoming friends of friends-of-friends, and so on. Two types of edges can be added to increase the In-Closeness centrality of the target node v : *long-haul edges* from far away ancestors at level L where $L > 3$ edges or *short-haul edges* from ancestors at level $L = 2$ to the target node v . To find the ancestors and their levels, we first use an algorithm (traversing based on Breadth-First Search) to get the predecessors' list, $L_{anc}(v)$, of the target node v and their levels (How far they are from v). Then, we assume node v has some information (e.g. In-Closeness centrality) about its ancestors either from the platform owner or by analyzing the paths associated with received requests and their traffic. Node v ranks its ancestors in each level on the basis of their In-Closeness centrality.

Experimentally, we find that (1) adding an edge, $e_{u,v}$, from an ancestor u at any level to the target node v will only increase v 's In-Closeness centrality (Other centrality measures might be effected, but this is not our goal in this study); (2) the most important nodes that dramatically increase the In-Closeness centrality when k is larger than 25, are friends of friends-of-friends at the level-3 of node v . As shown in Table 3, if we compare the increase in In-Closeness centrality after connecting a random number (150 edges in the shown experiment) of ancestors in each level to the target node v (nodes id: 764, 707, 19, 342, and 1535 of dataset DS-6) separately; (3), the maximum number of levels for most nodes is six. This is compatible with the six degrees of separation.

Each node in OSN has a different strategy and this strategy changes with time. Some Nodes sometimes need to increase their Out-Closeness centrality to be close to their targets and some other times they do not need to. Other nodes sometimes need to increase their In-Closeness centrality to be close to their sources and some other times they do not need to. It is the node's responsibility to make the trade-off between its desire to change centrality and the extra load/traffic which it might get. A node can use two techniques to increase its In-Closeness centrality: the *Adding* technique or the *Replacement* technique. The drawback of the Adding technique is that a target node after a period of

Table 3. The change of In-Closeness centrality of different nodes at different levels

Level/Node-ID	764	707	19	342	1535
<i>Baseline</i>	0.287	0.274	0.298	0.245	0.192
Level-2	0.322	0.314	0.329	0.308	0.274
Level-3	0.344	0.335	0.351	0.310	0.293
Level-4	0.316	0.312	0.328	0.287	0.299
Level-5	0.292	0.278	0.301	0.257	0.265
Level-6	0.289	0.275	0.298	0.247	0.200

time will be flooded with friends. The other option is to replace the edge (delete the existing edge before adding). A node is given k edges to add, but it must delete k (more or less based on equation) directly connected edges. Based on the finding number 2, a set of algorithms and techniques can be introduced to focus on friends of friends-of-friends at level-3 and ignore other levels.

6 Conclusion

In this paper, we experimentally find that (1) nodes with high Out-Closeness centrality in OSN suffer from high end-to-end delay as a target, but not as a source, (2) the cause of this end-to-end delay is that most nodes with high Out-Closeness centrality have low In-Closeness centrality, (3) the best level to increase the In-Closeness centrality of a node is its Friends of Friends-Of-Friends (Level-3). Moreover, we show that the increase in the local In-Degree centrality increases the global In-Closeness centrality. The Social Priority based Path Delay (SPPD) Metric is used for estimating the end-to-end social routing delay. An agent-based Model for Social Routing is proposed and a set of large-scale Google+ Graphs are used. We conducted a simulation study to compare the average end-to-end delays from the source and target perspectives by propagating a set of requests in four different societies with different routing schemes and diverse queue disciplines.

References

1. Barabasi, A.-L.: The origin of bursts and heavy tails in human dynamics. *Nature* **435**(7039), 207–211 (2005)
2. Borgatti, S.P.: Centrality and network flow. *Soc. Netw.* **27**(1), 55–71 (2005)
3. Brodka, P., Musial, K., Kazienko, P.: Efficiency of node position calculation in social networks. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009, Part II. LNCS, vol. 5712, pp. 455–463. Springer, Heidelberg (2009)
4. Carrington, P.J., Scott, J., Wasserman, S.: *Models and methods in social network analysis*, vol. 28. Cambridge University Press, Cambridge (2005)
5. Chapela, V., Criado, R., Moral, S., Romance, M.: *Intentional Risk Management Through Complex Networks Analysis*. Springer, Heidelberg (2015)

6. Cheng, Y.-Y., Lee, R.K.-W., Lim, E.-P., Zhu, F.: DelayFlow centrality for identifying critical nodes in transportation networks. In: 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE (2013)
7. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**(3), 215–239 (1979)
8. Ilyas, M.U., Radha, H.: Identifying influential nodes in online social networks using principal component centrality. In: 2011 IEEE International Conference on Communications (ICC). IEEE (2011)
9. Kang, U., Papadimitriou, S., Sun, J., Tong, H.: Centralities in large networks: algorithms and observations. In: Proceedings of the SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics (2011)
10. Karlin, J., Forrest, S., Rexford, J.: Nation-state routing: censorship, wiretapping, and BGP (2009). arXiv preprint [arXiv:0903.3218](https://arxiv.org/abs/0903.3218)
11. Kayes, I., Qian, X., Skvoretz, J., Iamnitchi, A.: How influential are you: detecting influential bloggers in a blogging community. In: Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C. (eds.) SocInfo 2012. LNCS, vol. 7710, pp. 29–42. Springer, Heidelberg (2012)
12. Kim, H., Tang, J., Anderson, R., Mascolo, C.: Centrality prediction in dynamic human contact networks. *Comput. Netw.* **56**(3), 983–996 (2012)
13. Klema, V.C., Laub, A.J.: The singular value decomposition: Its computation and some applications. *IEEE Trans. Autom. Control* **25**(2), 164–176 (1980)
14. Othman, S., Khan, J.I.: SOR: a protocol for requests dissemination in online social networks. In: Agarwal, N., Xu, K., Osgood, N. (eds.) SocInfo 2012. LNCS, vol. 9021, pp. 394–399. Springer, Heidelberg (2015)
15. Sporns, O., Honey, C.J., Kötter, R.: Identification and classification of hubs in brain networks. *PLoS ONE* **2**(10), e1049–e1049 (2007)
16. Varga, A.: The OMNeT++ discrete event simulation system. In: Proceedings of the European simulation Multi-conference (ESM 2001) (2001). SN