Distributed Assessment of the Closeness Centrality Ranking in Complex Networks

Klaus Wehmuth
National Laboratory for Scientific Computing (LNCC)
klaus@lncc.br

Artur Ziviani
National Laboratory for Scientific Computing (LNCC)
ziviani@lncc.br

ABSTRACT

We propose a method for the Distributed Assessment of the Closeness CEntrality Ranking (DACCER) in complex networks. DACCER computes centrality based only on localized information restricted to a given neighborhood around each node, thus not requiring full knowledge of the network topology. We show that the node centrality ranking computed by DACCER is highly correlated with the node ranking based on the traditional closeness centrality, which requires high computational costs and full knowledge of the network topology. This outcome is quite useful given the vast potential applicability of closeness centrality, which is seldom applied to large-scale networks due to its high computational costs. Results indicate that DACCER is simple, yet efficient, in assessing node centrality while allowing a distributed implementation that contributes to its performance. This also contributes to the practical applicability of DACCER in the analysis of large-scale complex networks, as we show using in our experimental evaluation both synthetically generated networks and traces of real-world networks of different kinds and scales.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]

General Terms

Design, Algorithms, Performance

Keywords

Network Science, Centrality, Distributed Algorithm, Closeness

1. INTRODUCTION

The concept of network centrality is an important tool to analyze complex networks [1, 11, 18]. In broad terms, network centrality measures the relative importance of nodes in a complex network. Different ways of measuring centrality have been proposed for decades [9, 20], each of them suited to assess node centrality from a different point of view. Examples include using network centrality to evaluate network robustness to fragmentation or to identify the most important nodes for efficient information spreading in diffusion networks.

As the definitions for centrality vary, so varies the difficulty in computing centrality, ranging from low cost (*e.g.*, degree centrality) to others far more costly, such as betweenness and closeness centralities. The later two, even though very useful, are costly because they rely on the determination of the shortest path between all pairs of nodes, thus also requiring full knowledge of the network topology. A high computational cost and the requirement of full knowledge of network topology becomes a significant obstacle for applying the general concept of network centrality to the large-scale complex communication networks we face nowadays, such as the Internet routing structure, online social networks, P2P networks, and content distribution networks. Hence, research in network science has been recently dedicated to dealing with centralities in large-scale networks (Section 4 reviews related work). Typically, these recent efforts either (i) optimize the way traditional centralities are calculated or approximated [4, 21]; or (ii) propose methods to distributively assess network centrality without requiring full knowledge of the network topology [10, 12, 17].

In this paper, we propose DACCER (*Distributed Assessment of the Closeness CEntrality Ranking*), a distributed method to assess network centrality based only on localized information restricted to a given neighborhood around each node. DACCER computes centrality in a fully distributed way, without requiring full knowledge of the network topology. In centrality-based network analysis, the position of each node in the centrality ranking is typically more important than the particular centrality value associated to each node. The node centrality ranking computed by DACCER is highly correlated with the node ranking based on the traditional closeness centrality, which requires high computational costs and full knowledge of the network topology. This outcome is quite useful given the vast potential applicability of closeness centrality, which is seldom applied to large-scale networks due to its high computational costs even if the full network topology is known. We show DACCER is simple, yet efficient, in distributively assessing network centrality. This conclusion stems from a thorough evaluation of DACCER using both synthetically generated networks and traces of real-world networks of different kinds and scales.

SIMPLEX '12, April 17 2012, Lyon, France Copyright 2012 ACM 978-1-4503-1238-7/12/04$10.00

This paper proceeds as follows. Section 2 introduces DACCER. Section 3 presents results from applying DACCER to a diverse set of synthetic and real-world networks. We analyze related work in Section 4. Finally, Section 5 concludes the paper and discusses future work.

2. DACCER

This section describes DACCER. Section 2.1 introduces key definitions and the theoretical ground to understand DACCER. Section 2.2 presents how the centralities are distributively computed. In Section 2.3, we discuss the main properties of DACCER.

2.1 Key definitions and background

We consider a network as equivalent to an *undirected* finite simple graph $G = (V, E)$, where V is the set of nodes and E the set of edges. The distance between two nodes in the network is defined as the number of hops in the shortest path connecting these nodes. The radius r of a graph $G = (V, E)$ is equivalent to the minimum eccentricity of any node, *i.e.*, $r = \min_{i \in V}(\max_{j \in V} d(i,j))$ where $d(i,j)$ is the shortest path distance between nodes i and j. In the following, we introduce key definitions for understanding DACCER in the next subsections.

Definition 1. *We define the Frobenius norm of a network with n nodes as the Frobenuis norm $||\mathbf{A}||_F$ of the adjacency matrix $\mathbf{A}_{n \times n} = \{a_{ij}\}$ of the network's associated graph. The Frobenius norm $||\mathbf{A}||_F$ is defined as*

$$||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2}. \quad (1)$$

Definition 2. *We define the h_neighborhood of a node i, where i={1,...,n}, as the object containing nodes with distance to node i less or equal to h and the edges that are adjacent to at least one of those nodes. The h_neighborhood of each node i is referred to as $H_h^i = (V_h^i, E_h^i)$, where V_h^i is the set of nodes with distance to node i less or equal to h and E_h^i is the set of edges adjacent to at least one of the nodes in V_h^i. Therefore, the H_0^i is the object where V_0^i contains just the node i and E_0^i contains the edges adjacent to i. Similarly, H_1^i is the object where V_1^i contains the node i and all its direct neighbors, while E_1^i contains the edges adjacent to at least one of those nodes. In H_2^i, V_2^i contains node i, its neighbors, and all the neighbors of its neighbors, while E_2^i contains the edges adjacent to at least one of the nodes in V_2^i. The number of nodes in H_h^i is represented as $|H_h^i|$.*

Definition 3. *We define the volume $Vol(H_h^i)$ of a neighborhood H_h^i as the sum of the degrees of its nodes, i.e.*

$$Vol(H_h^i) = \sum_{j \in H_h^i} d_j, \quad (2)$$

where d_j is the degree of node j. Note that this also includes all edges that connect nodes in H_h^i to nodes outside it. Clearly, $H_{h+1}^i \supseteq H_h^i$ and $Vol(H_{h+1}^i) \geq Vol(H_h^i)$.

Definition 4. *Given a network N and a neighborhood H_h^i around any node $i \in N$, we define the adjacency matrix of H_h^i with the same dimensions $n \times n$ of the adjacency matrix of the whole network N, containing 1s for the entries that represent edges present in H_h^i and 0 elsewhere. It should be clear from this definition that for every entry equal to 1 in the adjacency matrix of H_h^i there is an entry equal to 1 in the adjacency matrix of N, but the converse is not necessarily true.*

Theorem 1. *The volume of a network is the square of its Frobenius norm.*

Proof. *Let $G = (V, E)$ be the graph associated to the network N, V the set of nodes and E the set of edges. From the definitions of volume and Frobenius norm,*

$$Vol(N) = \sum_{i \in V} d_i = \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2 = (||\mathbf{A}||_F)^2. \quad (3)$$

Since all entries of an adjacency matrix of a undirected simple graph are either 0 or 1, squaring them has no effect. Thus, considering the well-known fact that for this kind of graph the sum of entries in a adjacency matrix equals its volume, we conclude that Eq. 3 holds as well as Theorem 1. □

2.2 Distributed assessment of centrality

In DACCER, the centrality value of each node i in the network is defined as the $Vol(H_h^i)$, as stated in Definition 3. Since Theorem 1 shows that the volume is the square of the Frobenius norm for a network, it becomes clear that the volume is a valid form to discriminate *h_neighborhood*s. It can be seen from this and from Definition 4 that two *h_neighborhood*s have the same volume if and only if they have the same Frobenius norm. In this sense, the neighborhood H_h^i can be viewed as an approximation to the whole network as h increases, since the Frobenius norm $||G - H_h^i||_F$ decreases when $Vol(H_h^i)$ increases and goes to zero when $H_h^i = G$. Also, given a determined radius (*e.g.* $h = 2$), the neighborhood with the highest volume will be the one more connected and therefore denser, leading to the idea that the central node i of such neighborhood will be closer to the other nodes, showing the basis for the correlation with closeness centrality ranking. Further, it can be seen that, for this correlation to hold, the neighborhoods should in a way be representative of the whole network and so, the radius of the network should not be much larger than the radius chosen for the neighborhoods. From this, it can be seen that when applied with small neighborhood radius (*e.g.* $h = 2$) this method should work for networks where the network's radius is relatively small as compared to the size of the network, such as networks with the small world property. This represents a large family of practical networks with many examples of real-world networks of interest.

To compute the volume-based centrality values for all the nodes in the network, we choose a value for h and then find $Vol(H_h^i)$ for each node i. Clearly, the choice of h directly impacts the obtained result. With $h = 0$, this localized centrality becomes the traditional degree centrality. With $h > 0$, each node i needs to discover its own H_h^i along with the degree of each node belonging to it. To achieve this, each node sends its identity and degree to each of its neighbors in a message with time-to-live (TTL) equal to h. This message also carries a unique message id (*e.g.*, the originator node id plus a time stamp) in order to prevent retransmissions of repeated messages. Upon receiving such a message, each node checks the message id to determine if it has received this message before. If the message is new, the node stores the provided information—since it is necessary for determining its own *h_neighborhood*. As only localized information is required, buffer complexity at each node i is limited to $O(|H_h^i|)$. The node then decrements the TTL. If the TTL is not zero, the node relays the message to all its neighbors; otherwise, no further action is taken. This runs in parallel at each node and after h steps, all nodes know their *h_neighborhoods* and the degree of its components.

We analyze the message and time complexity of this algorithm in the following. First consider the extreme case of h being sufficiently large to cover all the network, *i.e.,* the H_h^i for each node i encompasses every other node in the network. In this case, since every node only forwards new messages, an absolute upper bound for the number of messages equals the number of nodes times the total volume of the network. When applying DACCER to practical cases, however, an h significantly lower than the radius of the network (*e.g.,* $h = 2$) is enough to generate localized information able to achieve a suitable trade-off between efficiency in assessing network centrality and applicability costs, as we analyze later in Section 3.2. For $h = 2$, each node sends a message to all its neighbors and these neighbors in turn forward each new received message to their neighbors. Therefore, for $h = 2$, the expected message complexity is $O(n \times d_{avg}^2)$, where n is the number of nodes in the network and d_{avg} is the network average degree. As for the time complexity, the information generated at each node has to spread for h hops in order to reach all its destinations. Therefore, the expected time complexity is $O(1)$ steps once an h is chosen.

After each node i has determined its H_h^i neighborhood, the centrality values for all nodes can be calculated. For that, each node i calculates $Vol(H_h^i)$, as stated in Definition 3. Since all information needed for this computation is already known by each node, no message exchange is necessary. Further, each localized volume $Vol(H_h^i)$ consists of a simple sum of $|H_h^i|$ terms, therefore having a modest computational cost for each node.

2.3 DACCER properties

As we previously mentioned, in the trivial case where $h = 0$, the localized volume-based centrality exactly matches the traditional degree centrality. However, as h increases, the *h_neighborhood* with the largest volume in general coincides with the *h_neighborhood* with the largest number of nodes. Moreover, since the volume considers all the connections to nodes outside the neighborhood, this means that the *h_neighborhood* with the largest volume is associated to the $(h+1)$*_neighborhood* with the largest number of nodes. Intuitively, we observe that this construction is highly related to the concept of the traditional *closeness centrality*, since the closeness centrality can be defined in terms of how many nodes can be reached at increasing distances from the node in consideration. From these properties, we can intuitively expect the localized volume-based centrality provided by DACCER to correlate well with the traditional closeness centrality. This is indeed experimentally confirmed in Section 3.1. We can also expect the strength of this correlation to depend on the network topology as well as on the choice of h. Later in Section 3.2, we empirically show that for synthetic networks and trace-based real-world networks of different kinds—both in structure and scale—, $h = 2$ provides a suitable choice.

3. PERFORMANCE EVALUATION

In this section, we evaluate the performance of DACCER in assessing the network centrality in a distributed way. We present experimental results using different networks on (i) the correlation between rankings provided by DACCER and by closeness centrality; (ii) the trade-off between neighborhood size and applicability costs; (iii) the practical applicability of DACCER in large-scale networks.

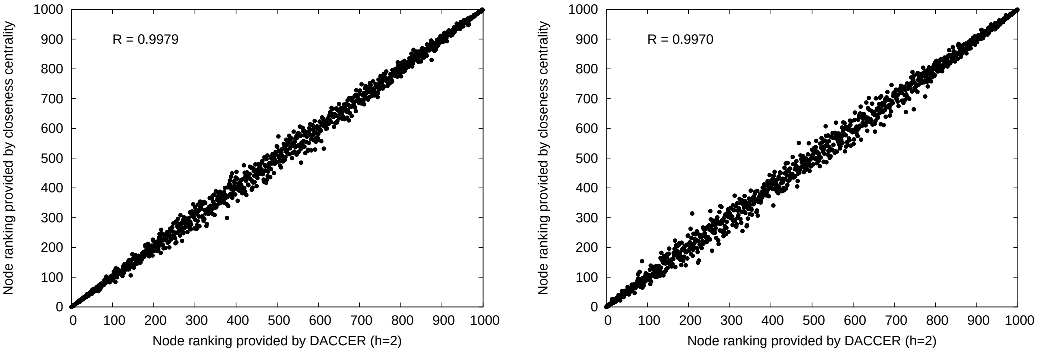
3.1 DACCER and closeness centrality

At the end of Section 2.2, we argue that we expect a high level of correlation between the node ranking provided by closeness centrality and the node ranking provided by DACCER. In this section, we experimentally confirm this claim by analyzing the correlation between these rankings obtained from different synthetically generated networks as well as traces of real-world networks.

We first evaluate the correlation obtained in two kinds of synthetic networks: 100 scale-free networks based on the Barabási-Albert (BA) model [2] and 100 random networks based on the Erdös-Rényi (ER) model [7]. The BA networks have 1,000 nodes each and are created with 5 connections per new node, resulting in a 9.95 mean node degree. The ER networks also have 1,000 nodes and a connection probability $p = 0.01$, which corresponds to $1.5 \times \frac{\ln 1000}{1000}$, ensuring that the resulting networks are connected as it is known that $p > \frac{\ln n}{n}$ is a sharp threshold for the connectedness of ER networks. The mean degree of the ER networks can vary, but in this case remains close to 10.

Figure 1 shows the correlation between the rankings provided by DACCER ($h = 2$) and by closeness centrality for each node in one BA and one ER network, both randomly chosen among the set of 100 BA and 100 ER networks. We observe a high correlation in both cases: for the BA network the correlation coefficient R between the rankings based on closeness centrality and DACCER ($h = 2$) is $R = 0.9979$ while for the ER network is $R = 0.9970$. Considering the whole set of 100 BA and 100 ER networks, all results for the correlation coefficient are between $R_{min} = 0.9972$ and $R_{max} = 0.9986$ for the BA networks and between $R_{min} = 0.9962$ and $R_{max} = 0.9975$ for the ER networks.

We next analyze the correlation between node rankings provided by closeness centrality and by DACCER using the network traces specified in Table 1. Actors represents a social network where edges link actors (nodes) who worked in the same movie [2]. Routers-CAIDA refers to the giant connected component of a router-level network topology collected by CAIDA [5]. RouteViews represents a symmetrized snapshot of the AS-level Internet structure reconstructed from BGP tables [19]. PGP-net refers to a network of users



(a) BA Network

(b) ER Network

Figure 1: Correlation between rankings provided by DACCER and closeness centrality.

Table 1: Real-world network traces and their ranking correlation for $h = 2$.

Network trace	# of nodes	# of edges	radius	ranking correlation (R)
Actors [2]	374,511	15,014,850	4	0.9537
Routers-CAIDA [5]	190,914	607,610	13	0.9066
RouteViews [19]	22,693	48,436	6	0.9954
PGP-net [3]	10,680	24,316	12	0.8704

of the Pretty-Good-Privacy algorithm for secure information exchange [3]. Limiting message complexity by setting $h = 2$, the correlation coefficient was high for all considered networks, as shown in Table 1. In Section 3.2, we perform a cost-effectiveness analysis that indicates $h = 2$ as a suitable choice to balance the trade-off between cost and the resulting correlation between rankings provided by closeness centrality and by DACCER.

The high correlation between the node rankings provided by DACCER and by closeness centrality constitutes a key outcome. Closeness centrality is a basic metric to analyze complex networks. To the best of our knowledge, however, there is no distributed method to compute the closeness centrality. Even if the full network topology is known, closeness centrality is too costly—$O(nm + n^2 \log n)$ where n is the number of nodes and m is the number of edges—to be applied in very large complex networks. DACCER thus provides a simple, efficient, and practical alternative to rank nodes in very large complex networks in close relation with the node ranking by closeness centrality.

3.2 Trade-off between h and message cost

An increase on h causes an increase in the number of messages needed to obtain the localized volume-based centrality of each node. Hence, one has to consider the cost-effectiveness relation of increasing h, balancing the trade-off between the message cost and the correlation coefficient of the node rankings provided by DACCER and by closeness centrality.

Figure 2 shows this trade-off for the traces of real-world networks (Table 1). The vertical axis at the left refers to the correlation coefficient of the node ranking provided by DACCER and the one provided by closeness centrality. The vertical axis at the right shows the normalized number of messages. The normalization is done for the sake of comparability so that the message cost for $h = 4$ equals one, and all other cost are proportional to this. For all three networks the best trade-off between the correlation coefficient and the message cost happens with $h = 2$—*i.e.*, the message cost is still low and the correlation coefficient is relatively high. The same is also valid for all the synthetically generated networks considered in this paper. This suggests $h = 2$ provides a suitable cost-effectiveness balance.

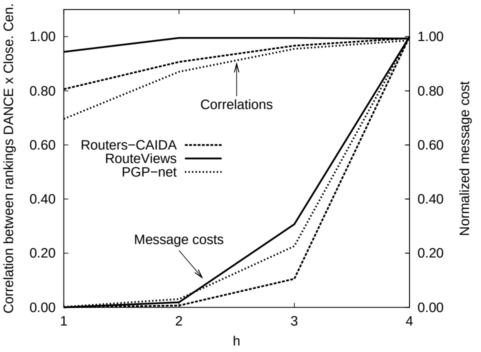
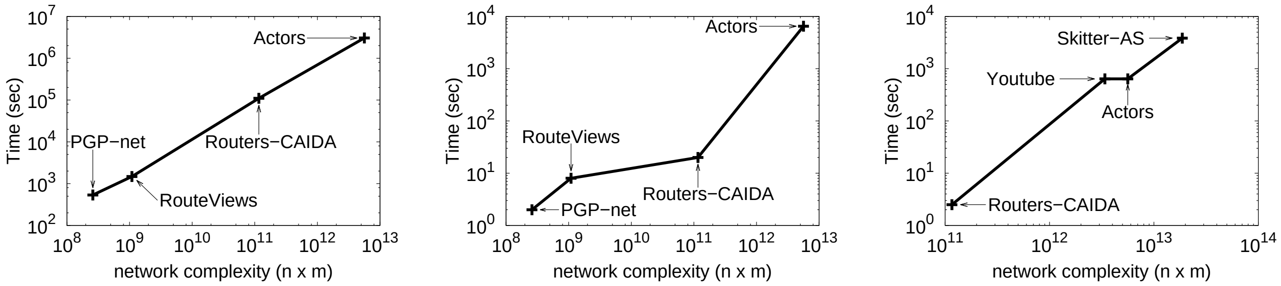


Figure 2: Trade-off between the message cost and the correlation coefficient between rankings by DACCER and closeness centrality with increasing h.

3.3 Practical applicability of DACCER

Computing the localized volume-based centrality with DACCER only requires each node to know the degrees of the nodes belonging to its *h_neighborhood*. This means DACCER can be used in networks where the topology is fully known and also in networks where each node only knows



(a) Closeness 1 CPU (b) DACCER 1 CPU (c) DACCER 32 CPUs

Figure 3: Execution times for closeness and DACCER node rankings.

its direct neighbors. Furthermore, DACCER can be implemented in different ways, ranging from a centralized approach running on a single CPU core to a fully distributed approach with the analysis of each node running on a separate core.

To evaluate the practical applicability of DACCER, we compare the time spent for calculating the closeness centrality ranking using the traditional algorithm running on a single CPU core with DACCER implementations running on 1 and 32 CPU cores, each core equivalent to the one used for traditional closeness. This study evaluates the performance gains allowed by DACCER, even in a modestly parallelized implementation, when compared with the traditional implementation of closeness centrality, which is not easily parallelizable. In the parallelized DACCER implementation, the network nodes are distributed among the 32 cores and each core calculates the centrality of its assigned nodes. The whole network topology information is available to all cores in shared memory.

Figure 3 presents the execution time in seconds to compute the closeness centrality ranking and the DACCER ranking as a function of the network complexity. As the measure of network complexity, we consider the number of nodes n times the number of edges m of each analyzed network. Figures 3(a) and 3(b) show the time spent to compute the closeness centrality ranking and the DACCER ranking for the network traces presented in Table 1 in a single CPU core, respectively. In turn, in Figure 3(c), we present the time spent to compute the centrality ranking based on DACCER using 32 CPU cores. Figure 3(c) shows results for the two largest networks in Table 1, namely Routers-CAIDA and Actors, and for two additional large networks: the anonymized social network of Youtube users [16] with 1,134,890 nodes and 2,987,624 edges and the Internet's Autonomous systems network by Skitter [13] with 1,696,415 nodes and 11,095,298 edges. These two latter networks present a scale that strongly limits the practical applicability of the closeness centrality to rank their nodes. We emphasize that Figure 3(a) uses a different scale for the execution time than Figures 3(b) and 3(c). For instance, the computation of the closeness centrality ranking for the Actors network takes 3×10^{6} seconds (roughly 34 days) in the single CPU core, whereas the DACCER centrality ranking is computed after 6,500 seconds (about 1,8 hours) using the single CPU core and in only 640 seconds (about 10 minutes) using the 32 CPU cores. From this experiment, it becomes clear that, even with a modest parallelization, DACCER is orders of magnitude faster than the traditional closeness centrality algorithm. Therefore, the possibility of parallelizing DACCER execution makes it applicable to the analysis of large-scale networks for which it would be unfeasible in practice to compute closeness centrality. For instance, the Skitter-AS network is analyzed using DACCER in about 1 hour using the 32 CPU cores. Further, DACCER does not require full knowledge of the network topology, being therefore applicable on networks where the topology is not fully known.

4. RELATED WORK

There are many centrality measures for assessing the relative importance of nodes in a network under different criteria, such as the capacity for information diffusion or relevance for connectivity. Examples are the traditional degree, betweenness, closeness, and eigenvector centralities [9, 20].

The computing of most of the traditional centralities is in general computationally expensive and requires full knowledge of the network topology. Therefore, some recent efforts are dedicated to optimize the way by which traditional centralities are calculated or approximated [4, 21]. These methods, however, still require full knowledge of the network topology to compute a centrality approximation, hindering their applicability to large-scale networks where such an information is unavailable and a distributed implementation is required.

Alternatively, as our proposal, some previous works investigate methods to assess network centrality in a distributed way, without requiring full knowledge of the network topology [10, 12, 17]. Lehmann and Kaufmann [12] propose a framework for computing shortest-path based centralities, such as closeness and betweenness, in a decentralized way, but their proposal is still computationally expensive for application to large-scale complex networks. Kermarrec et al. [10] use a random walk to distributively assess network centrality in complex networks, however their random walk approach does not particularly correlate with closeness centrality and presents a high convergence time. Lim et al. [14] find the top-k centrality nodes on a network by sampling. Ercsey-Ravasz and Toroczkai [6] approximate betweenness centrality using only local neighborhoods on a network. Nanda and Kotz [17] propose a new centrality metric called Localized Bridging Centrality (LBC). LBC provides a specialized centrality targeted at locating bridges, *i.e.*, edges whose removal disconnects the network using only one-hop neighborhoods around each node. The proposed use of LBC is on relatively small-scale wireless mesh networks. One of the main motivations behind the LBC proposal is a paper by Marsden [15], which shows empirical evidence that localized centrality measures computed for one-hop neighborhood are highly correlated to a global centrality measure. Everett and Borgatti [8] explore this notion to approximate betweenness centrality. In this paper, we extend this notion by proposing DACCER and showing that the node ranking based on its localized volume-based centrality correlates well with the closeness centrality ranking. In a previous work [22], we applied a similar method using spectral analysis to locally assess centrality, but this approach does not correlate with closeness centrality and present a higher computational cost than our current proposal.

5. CONCLUSION

In this paper, we propose DACCER, a novel distributed method to approximate the closeness centrality ranking in large complex networks, without requiring full knowledge of the network topology. DACCER computes a localized volume-based centrality at each node considering only a limited neighborhood around every node. In short, DACCER achieves a node ranking that is highly correlated with the ranking based on the traditional closeness centrality, whereas with applicability costs that are significantly lower. We show $h = 2$ presents a suitable trade-off between limited message costs and high correlation with the closeness centrality ranking. This depends on the network radius not being too large compared to h. Most complex networks of interest present *small world* properties (*i.e.*, small radius compared to network size), thus rendering DACCER applicable to the practical analysis of these networks. Overall, DACCER contributes with a simple yet efficient method to approximate closeness centrality ranking in large-scale complex networks that can be run in a fully distributed way.

Most complex networks also present dynamic behavior. As future work, we plan to investigate how DACCER can contribute to the analysis and modeling of dynamic networks.

6. ACKNOWLEDGMENTS

This work was partially supported by the Brazilian Funding Agencies FAPERJ and CNPq. Authors thank Alan Mislove (Northeastern University) for providing the datasets used in [16]. Authors also thank Ana Paula C. da Silva (UFJF), Antônio Tadeu A. Gomes (LNCC), and Daniel R. Figueiredo (UFRJ) for their comments on our work.

7. REFERENCES

[1] R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47–97, 2002.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] M. Boguna, R. Pastor-Satorras, A. Diaz-Guilera, and A. Arenas. List of edges of the giant component of the network of users of the Pretty-Good-Privacy algorithm for secure information interchange. *Physical Review E*, 70(056122), 2004.

[4] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[5] CAIDA. CAIDA's router-level topology measurements, 2003.

[6] M. Ercsey-Ravasz and Z. Toroczkai. Centrality scaling in large networks. *Physical Review Letters*, 105(3):038701, 2010.

[7] P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[8] M. Everett and S. Borgatti. Ego network betweenness. *Social Networks*, 27(1):31–38, 2005.

[9] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[10] A.-M. Kermarrec, E. Le Merrer, B. Sericola, and G. Trédan. Second order centrality: Distributed assessment of nodes criticity in complex networks. *Computer Communications*, 34(5):619–628, Apr. 2011.

[11] H. Kim, J. Tang, R. J. Anderson, and C. Mascolo. Centrality prediction in dynamic human contact networks. *Computer Networks*, 2012. To appear.

[12] K. Lehmann and M. Kaufmann. Decentralized algorithms for evaluating centrality in complex networks. Technical report, Wilhelm Schickard Institut, 2003.

[13] J. Leskovec, J. Kleinber, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005.

[14] Y.-S. Lim, D. S. Menasche, B. Ribeiro, D. Towsley, and P. Basu. Online estimating the k central nodes of a network. In *Proc. of the IEEE Network Science Workshop (NSW)*, June 2011.

[15] P. Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407–422, Oct. 2002.

[16] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proc. of the ACM/Usenix Internet Measurement Conference (IMC)*, San Diego, CA, Oct. 2007.

[17] S. Nanda and D. Kotz. Localized bridging centrality for distributed network analysis. In *Proc. of Int. Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, Aug. 2008.

[18] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[19] M. E. J. Newman. Internet – a symmetrized snapshot of the structure of the internet at the level of autonomous systems, 2006.

[20] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[21] Z. Shi and B. Zhang. Fast network centrality analysis using GPUs. *BMC Bioinformatics*, 12(1):149, 2011.

[22] K. Wehmuth and A. Ziviani. Distributed algorithm to locate critical nodes to network robustness based on spectral analysis. In *Proc. of the Latin American Network Operations and Management Symposium (LANOMS)*, Oct. 2011.