

## Temporal node centrality in complex networks

Hyounghick Kim and Ross Anderson

*Computer Laboratory, University of Cambridge, 15 J. J. Thomson Avenue, Cambridge CB3 0FD, United Kingdom*

(Received 1 November 2011; published 13 February 2012)

Many networks are dynamic in that their topology changes rapidly—on the same time scale as the communications of interest between network nodes. Examples are the human contact networks involved in the transmission of disease, *ad hoc* radio networks between moving vehicles, and the transactions between principals in a market. While we have good models of static networks, so far these have been lacking for the dynamic case. In this paper we present a simple but powerful model, the *time-ordered graph*, which reduces a dynamic network to a static network with directed flows. This enables us to extend network properties such as vertex degree, closeness, and betweenness centrality metrics in a very natural way to the dynamic case. We then demonstrate how our model applies to a number of interesting edge cases, such as where the network connectivity depends on a small number of highly mobile vertices or edges, and show that our centrality definition allows us to track the evolution of connectivity. Finally we apply our model and techniques to two real-world dynamic graphs of human contact networks and then discuss the implication of temporal centrality metrics in the real world.

DOI: [10.1103/PhysRevE.85.026107](https://doi.org/10.1103/PhysRevE.85.026107)

PACS number(s): 89.75.Hc

### I. INTRODUCTION

Many important phenomena depend on networks, from the spread of disease in a population through systems of metabolic processes to explicit networks such as the internet and the worldwide web. Recent advances in the theory of networks have provided us with the mathematical and computational tools to understand them better [1]. Often the topology of a network has distinctive features, such as vertex order distribution, clustering, and characteristic path length, which can be explained in terms of its evolution and which in turn explain some aspects of its behavior. For example, networks that grow by preferential attachment may acquire a power-law distribution of vertex order which in turn makes them robust against random node failure—yet vulnerable to attacks targeted on high-degree nodes [2]. Insights like this can inform activities from public health to counterterrorism [3].

So most analyses and models have assumed that networks are static, typically represented in graph form as a number of nodes connected by edges. However, in real life many networks are dynamic. New nodes are added to the graph, some existing ones are removed, and edges come and go too. While researchers have studied these mechanisms as a means of explaining graph topology, the effects of dynamic topology have generally been ignored when considering how topology affects connectivity. Yet there are important networks whose topology changes rapidly, and its dynamic aspects have a significant effect on connectivity:

(a) In epidemiology, some possibly infective contacts between individuals are long term (friends, family) but many are fleeting (people in the street or the market place). Their relative importance may vary. In medieval times, infectious disease may have been largely transmitted by a small number of merchants traveling between markets in otherwise largely isolated towns, while in a modern urban society the super-spreading node may be a school. When faced with an epidemic, it is important to know whether you should impose travel restrictions or close schools.

(b) There is interest in *ad hoc* radio networks set up between moving vehicles to transmit information about congestion

and to provide emergency communications. Here, oncoming vehicles offer a shorter interaction time, but more rapid information dissemination, than vehicles going in the same direction.

(c) In military communications systems, nodes that act as local exchanges or that provide long-distance backhaul may become conspicuous because of the volumes of traffic they handle, even if the opponent cannot decrypt and understand it, so they may be targeted. So nodes may take turns; a new exchange may be selected frequently and at random.

Thus far, the models and analytical tools used to characterize dynamic network behavior have been somewhat limited. It is common to look at static snapshots of the network independently, or to average their characteristics [4]; for example, some researchers estimate a node's topological importance using the average value of its centrality over all static snapshots. Such analyses, however, are limited since they neglect temporal paths that cross over multiple snapshots that individually contain only partial paths. Tang *et al.* [5] proposed a method to identify important nodes using temporal versions of conventional centrality metrics (e.g., *closeness* and *betweenness*).

In this paper we extend their work to a more general and more realistic model. We present the *time-ordered graph*, which reduces a dynamic network to a static network with directed flows. This enables us not only to use the algorithms developed for static graphs, but also to define better metrics for dynamic graphs. We propose temporal centrality metrics based on the time-ordered graph and demonstrate their robustness and usefulness by applying them to a number of interesting edge and human contact networks, respectively.

### II. RELATED WORK

Traditional network analysis uses static networks, or models that aggregate node interaction during a time interval. Such models break down when the network topology changes fast enough. For example, aggregate models can underestimate path length since they ignore the time delays needed to construct paths.

A number of papers in recent years have tried to overcome such limitations. Kempe *et al.* [6] proposed a model of temporal networks as static graphs where every edge is labeled with the time the interaction took place. Ferreira [7] also views a dynamic network as a sequence of static graphs and seeks to tackle the fundamental network problems such as routing metrics, connectivity, and spanning trees for dynamic networks. Kostakos [8] independently presented the concept of temporal graphs and set out to compute the shortest path between nodes in an evolving network. Tang *et al.* [9] tried to develop a more general model by introducing a variable representing the speed at which a message travels. Most recently, Tang *et al.* [5] proposed temporal centrality metrics based on temporal paths in order to measure the importance of a node in a dynamic network. They investigated time-aware central node identification methods with several real data sets but the performance of the proposed metrics may be lower in practice since they assume *a priori* knowledge of each node’s future contacts. For example, Tang *et al.* [10] proposed an application based on the assumption that global knowledge of all past and future contacts is available.

Its focus is to design a simple and general model that gives a concise and general formulation of the temporal properties of every dynamic network. Our central idea is to model a dynamic network as a time-ordered digraph which pastes together its temporal snapshots with directed edges that link each node to its successor in time. This transforms any dynamic network into a larger but more easily analyzable static one. The transformation enables us to find accurate and robust generalizations of existing static network concepts.

**III. MODEL**

In this study, we assume that the time during which a network is observed is finite, from the start time  $t_{\text{start}}$  until the end time  $t_{\text{end}}$ . Without loss of generality, we set  $t_{\text{start}} = 0$  and  $t_{\text{end}} = T$ . A *dynamic network*  $G_{0,T}^D = (V, E_{0,T})$  on a time interval  $[0, T]$  consists of a set of vertices  $V$  and a set of temporal edges  $E_{0,T}$  where a temporal edge  $(u, v)_{i,j} \in E_{0,T}$  exists between vertices  $u$  and  $v$  on a time interval  $[i, j]$  such that  $i \leq T$  and  $j \geq 0$ . In the dynamic network the set of vertices  $V$  is always the same while the set of existing edges can be changed over time.

Most characterizations of dynamic networks discretize time by converting temporal information into a sequence of  $n$  network “snapshots.” We use  $w$  to denote the time duration of each snapshot (or time window size),  $T/n$ , expressed in some time unit (e.g., seconds or hours). In other words, a dynamic network can be represented as a series of static graphs  $G_1, G_2, \dots, G_n$ . The notation  $G_t$  ( $1 \leq t \leq n$ ) represents the aggregate graph which consists of a set of vertices  $V$  and a set of edges  $E_t$  where an edge  $(u, v) \in E_t$  exists only if a temporal edge  $(u, v)_{i,j} \in E_{0,T}$  exists between vertices  $u$  and  $v$  on a time interval  $[i, j]$  such that  $i \leq wt$  and  $j > w(t - 1)$ . In other words,  $G_t$  is the  $t$ th temporal snapshot of the dynamic network  $G_{0,T}^D$  during the  $t$ th time window. For simplicity, we here assume that  $w = 1$ .

For clarity, we introduce the following illustrative example. When  $T = 3$ , the dynamic network with the set of temporal edges in Table I can be represented as the aggregated graph

TABLE I. Example contacts in a dynamic network.

Edge	Time interval
(A, C)	[1,1]
(A, D)	[2,2]
(B, D)	[2,3]
(C, D)	[3,3]

where all edges are aggregated into a single graph  $G_{1,3}^S$  or the series of static networks  $G_1, G_2$ , and  $G_3$  as we explained. The visual representations are shown in Fig. 1. Unlike the aggregated view of the graph  $G_{1,3}^S$  in Fig. 1 (left), the series of static networks  $G_1, G_2$ , and  $G_3$  in Fig. 1 (right) shows the temporal relationships effectively.

Although this time series representation of the graph [see Fig. 1(right)] is intuitive, it is not easy to directly analyze the temporal characteristics of a dynamic network from its component snapshots. For example, when an edge  $(u, v)$  in a dynamic network represents the communication channel  $u$  and  $v$ , we may want to find the shortest possible route from  $u$  to  $v$ . In order to find a path from  $A$  to  $B$ , we have to wait at  $t = 0$ , use the path from  $A$  to  $D$  at  $t = 1$ , and then use the path from  $D$  to  $B$  at  $t = 2$ . How can we find this solution more generally?

We now construct the time-ordered graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  as the asymmetric directed graph shown in Fig. 2. Without loss of generality, we assume that the message transmission time is the same as  $w$ . In other words, at each time step, we can deliver a message along a single edge. It has a vertex  $v_t$  for each  $v \in V$  and for each  $t \in \{0, 1, \dots, n\}$ ; it has edges from  $u_{t-1}$  to  $v_t$  and vice versa for an edge  $(u, v) \in E[t]$ ; and it has edges from  $v_{t-1}$  to  $v_t$  for all  $v \in V$  and for all  $t \in \{1, \dots, n\}$ .

The point of this construction is that for every path between two nodes for given start and end times in a dynamic network, there is exactly one path between the corresponding vertices in the corresponding time-ordered graph, which thus captures all the connectivity information in the network. It has much finer granularity than any existing model [8,9] which assumes that a message can be delivered to the nodes within  $h$  hops at the same window. Such models are just approximations to a dynamic network; the time-ordered graph which we present here is a complete representation.

Its value is that it enables us to extend conventional graph theory algorithms to dynamic networks. For example, given a time-ordered graph  $\mathcal{G}$ , a *temporal shortest path* from node  $u$  to node  $v$  on a time interval  $[i, j]$  where  $0 \leq i < j \leq n$  is defined as any path  $p = \langle u_i, \dots, v_k \rangle$  where  $i < k \leq j$  with the path length  $|p| = \min_{i < l \leq j} \delta(u_i, v_l)$ , where  $\delta(u, v)$  is the shortest path distance from  $u$  to  $v$  in a static graph. Thus in

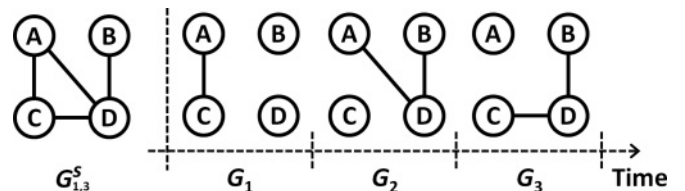


FIG. 1. Comparison of aggregated graph representation (left) and time series representation (right) of the contacts in Table I.

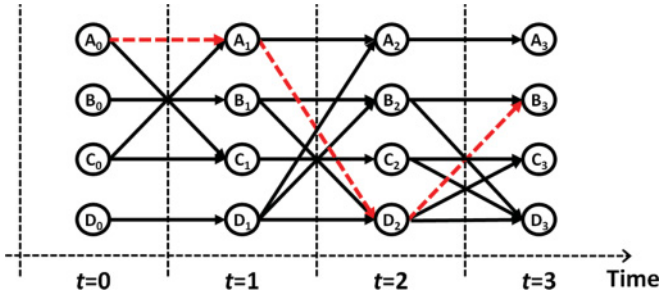


FIG. 2. (Color online) The corresponding time-ordered graph  $\mathcal{G}$  of Table I. The path consisting of dashed red edges represents a temporal shortest path from  $A$  to  $B$  on the time interval  $[0,3]$ .

Fig. 2 a temporal shortest path from  $A$  to  $B$  on the time interval  $[0,3]$  is clearly  $A_0, A_1, D_2$ , and  $B_3$ .

We will now develop definitions of centrality metrics (degree, closeness, betweenness, etc.) that capture the temporal characteristics of dynamic networks.

### A. Temporal degree

First, we need to generalize the concept of vertex degree to dynamic networks. We construct the temporal degree  $\mathcal{D}_{i,j}(v)$  for a node  $v \in V$  on a time interval  $[i, j]$  where  $0 \leq i < j \leq n$  as the normalized total number of inbound edges to and outbound edges from  $v$  on the time interval  $[i, j]$ , disregarding the “self-edges” from  $v_{t-1}$  to  $v_t$  for all  $t \in \{i+1, \dots, j\}$ . Now if our dynamic network is actually static, this sum is equal to  $\sum_{t=i}^j 2\mathcal{D}_t(v)$ , where  $\mathcal{D}_t(v)$  is the degree of  $v$  in  $G_t$ , so to normalize it we divide by  $2(|V| - 1)m$ , where  $m = j - i$ . Now a node’s normalized temporal degree is the same as the average value of the node’s degree in the time series of snapshot graphs.

Given a time-ordered graph  $\mathcal{G}$  derived from  $G_{i,j}^D = (V, E_{i,j})$ , the temporal degree values of all nodes in  $V$  can be computed in  $O(|V| + |\mathcal{E}|)$  time by checking the nodes adjacent to each edge in  $\mathcal{E}$ .

### B. Temporal closeness

A more difficult concept is temporal closeness, which we define as follows. The temporal closeness  $\mathcal{C}_{i,j}(v)$  for a node  $v \in V$  on a time interval  $[i, j]$  where  $0 \leq i < j \leq n$  is the sum of inverse temporal shortest path distances to all other nodes in  $V \setminus v$  for each time interval in  $\{[t, j] : i \leq t < j\}$ .

More precisely, we define temporal closeness by considering  $m$  time intervals  $\{[t, j] : i \leq t < j\}$  where  $m = j - i$  by varying the starting time  $t$  of each time interval from  $i$  to  $j - 1$  instead of one time interval  $[i, j]$  with the starting time  $i$ . We note that the time interval  $[i, j]$  contributes the *temporal shortest paths* only when the starting time is  $i$ ; the temporal shortest paths from node  $u$  to node  $v$  mean the paths from node  $u_i$  to node  $v_k$ , which is the first node encountered along a path from  $u_i$  to a node in  $\{v_{i+1}, \dots, v_j\}$ . However, the temporal shortest paths from  $u$  to  $v$  will change as time increases. Therefore, in addition to the case with the starting time  $i$ , we also need to consider the temporal shortest paths from node  $u$  to node  $v$  on the additional  $m - 1$  time intervals  $\{[t, j] : i < t < j\}$  by varying  $t$  from  $i + 1$  to  $j - 1$  to analyze

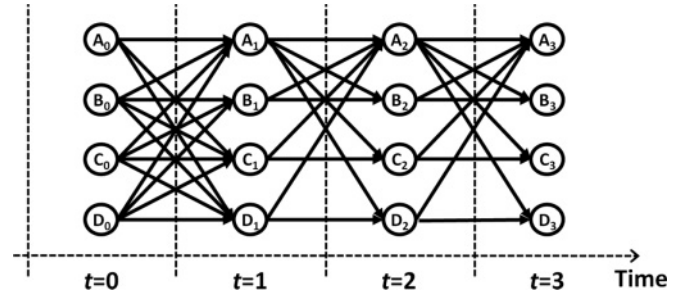


FIG. 3. An example of a time-ordered graph  $\mathcal{G}$  to explain our design philosophy for temporal closeness and betweenness. If we consider only the time interval  $[i, j]$ , the temporal shortest paths between all nodes are determined during the time interval  $[0, 1]$  alone regardless of subsequent changes, which does not help us measure the dynamic characteristics of this graph.

dynamic characteristics of the temporal shortest paths between  $u$  and  $v$  in a more reasonable manner.

The example in Fig. 3 illustrates our design principle. If we consider only the time interval  $[i, j]$  ( $[0,3]$  here), the temporal closeness values of all nodes are identical since all temporal shortest paths are determined during the time interval  $[0, 1]$  when the graph is fully connected; the subsequent interactions will be ignored in the computation. This is not satisfactory, as node  $A$  is clearly much more highly connected in the network than the other nodes. A reasonable temporal centrality metric should capture such dynamics. In the existing work [5], however, temporal metrics are defined with the time interval  $[i, j]$  alone rather than all the time intervals  $\{[t, j] : i \leq t < j\}$ .

Formally, the temporal closeness for a node  $v$  is

$$\mathcal{C}_{i,j}(v) = \sum_{i \leq t < j} \sum_{u \in V \setminus v} \frac{1}{\Delta_{t,j}(v,u)},$$

where  $\Delta_{t,j}(v,u)$  is the temporal shortest path distance from  $v$  to  $u$  on a time interval  $[t, j]$ . If there is no temporal path from  $v$  to  $u$  on a time interval  $[t, j]$ ,  $\Delta_{t,j}(v,u)$  is defined as  $\infty$ . Also, we note that  $\Delta_{t,j}(v,u)$  is different from  $\Delta_{t,j}(u,v)$  since the time-ordered graph  $\mathcal{G}$  is a directed graph.

In order to cover the cases when  $\Delta(v,u)$  is infinite, we use a slightly modified definition for closeness which is similar to the definition proposed by Opsahl *et al.* [11] for disconnected graphs. Here we assume  $1/\infty = 0$ . The temporal closeness is normalized by dividing each closeness value by  $(|V| - 1)m$  where  $m = j - i$ .

Given a time-ordered graph  $\mathcal{G}$  derived from  $G_{i,j}^D = (V, E_{i,j})$ , all-pair temporal shortest path distances can be computed in  $O(m|V|^2)$  time by using dynamic programming with the recurrence  $\Delta_{t,j}(v,u) = \Delta_{t+1,j}(k,u) + 1$  if  $(v,k) \in \mathcal{E}$ ; otherwise,  $\Delta_{t,j}(v,u) = 0$ . With the computed temporal shortest path distances, the temporal closeness value  $\mathcal{C}_{i,j}(v)$  of a node  $v$  in  $V$  can be computed in  $O(m|V|)$ , and thus the total running time of the temporal closeness computation for all nodes in  $V$  is  $O(m|V|^2)$ .

### C. Temporal betweenness

The betweenness centrality of a node is the proportion of shortest paths passing through it, so temporal betweenness



$\mathcal{B}_{i,j}(v)$  for a node  $v \in V$  on a time interval  $[i, j]$ ,  $0 \leq i < j \leq n$ , should be the sum of the proportion of all the temporal shortest paths through the vertex  $v$  to the total number of temporal shortest paths over all pairs of nodes for each time interval in  $\{[t, j] : i \leq t < j\}$ . For the same reason as in the temporal closeness definition, we consider  $m$  time intervals  $\{[t, j] : i \leq t < j\}$  where  $m = j - i$  instead of one time interval  $[i, j]$ .

Let  $\mathcal{S}_{x,y}(u, v)$  denote the set of temporal shortest paths from source  $s$  to destination  $d$  on the time interval  $[x, y]$  and  $\mathcal{S}_{x,y}(s, d, v)$  the subset of  $\mathcal{S}_{x,y}(s, d)$  consisting of paths that have  $v$  in their interior. Then, the temporal betweenness for a node  $v$  is

$$\mathcal{B}_{i,j}(v) = \sum_{i \leq t < j} \sum_{\substack{s \neq v \neq d \in V \\ \sigma_{t,j}(s, d) > 0}} \frac{\sigma_{t,j}(s, d, v)}{\sigma_{t,j}(s, d)},$$

where  $\sigma_{t,j}(s, d) \equiv |\mathcal{S}_{t,j}(s, d)|$  and  $\sigma_{t,j}(s, d, v) \equiv |\mathcal{S}_{t,j}(s, d, v)|$ . Temporal betweenness is normalized by dividing each betweenness value by  $(V_s^v V_d^v m)$  where  $m = j - i$  and  $V_s^v, V_d^v \subseteq V \setminus v$ , such that  $\sigma_{t,j}(s, d) > 0$  for each  $s \in V_s^v$ , for each  $d \in V_d^v$ , and for  $i \leq t < j$ .

Given a time-ordered graph  $\mathcal{G}$  derived from  $G_{i,j}^D = (V, E_{i,j})$ , the temporal betweenness values for all nodes in  $V$  can be efficiently calculated by using dynamic programming: For each node  $v \in \mathcal{V}$ ,  $\sigma_{t,j}(s, d, v) = \sigma_{t,k}(s, v) \sigma_{k,j}(v, d)$  if  $\Delta_{t,j}(s, d) = \Delta_{t,k}(s, v) + \Delta_{k,j}(v, d)$  where  $s \neq d \in V$  and  $i \leq t < k < j$ . Since  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is a directed acyclic graph, we compute  $\Delta_{t,j}(v, v)$  and  $\sigma_{t,j}(v, v)$  between  $v \in \mathcal{V}$  and  $v \in V$  where  $i \leq t < j$  in  $O(m^2|V|^2 + |\mathcal{E}|)$  time. We note that the worst-case running time of this computation is  $O(m^2|V|^2)$  since  $|\mathcal{E}|$  is  $O(m|V|^2)$  when all nodes are always completely connected to all nodes at each time step  $t$  where  $i \leq t < j$ . With the computed  $\Delta$  and  $\sigma$  values, for all nodes  $v \in \mathcal{V}$ , we can compute  $\sigma_{t,j}(s, d, v)$  in  $O(m^3|V|^3)$  time for  $i \leq t < j$ , and thus the total running time of the temporal betweenness computation is  $O(m^3|V|^3)$ . This algorithm requires  $O(m^2|V|^2)$  space to store  $\Delta$  and  $\sigma$  values.

#### D. Analysis of the example

As a sanity check, we compute our proposed metrics for the time-ordered graph in Fig. 2. For comparison, we also compute each node's centrality value in the aggregated graph [see Fig. 1 (left)] and the average centrality values in  $G_1$ ,  $G_2$ , and  $G_3$  [see Fig. 1 (right)]. The results are shown in Table II. From this table, we can see that  $A$  plays a relatively important role compared with  $C$  in the dynamic network while the network centrality values of  $A$  and  $C$  for the aggregated representation and the average centrality value in  $G_1$ ,  $G_2$ , and  $G_3$  are exactly identical. Intuitively, our metric seems reasonable as the edge  $(A, D)$  exists only at time  $t = 2$ .

In the next two sections we exercise our temporal metrics with more interesting and realistic test cases.

#### IV. WHY ARE TEMPORAL METRICS REALLY NECESSARY?

First, we define a dynamic network model which we call the *traveling merchant graph* (TMG) to model disease transmis-

TABLE II. Comparison of present results with static and average centrality metrics.

Node	Type	Degree	Closeness	Betweenness
A	<b>Temporal</b>	<b>0.222</b>	<b>0.426</b>	<b>0.133</b>
	Aggregated	0.667	0.750	0.000
	Average	0.222	0.259	0.000
B	<b>Temporal</b>	<b>0.222</b>	<b>0.370</b>	<b>0.000</b>
	Aggregated	0.333	0.600	0.000
	Average	0.222	0.296	0.000
C	<b>Temporal</b>	<b>0.222</b>	<b>0.370</b>	<b>0.000</b>
	Aggregated	0.667	0.750	0.000
	Average	0.222	0.259	0.000
D	<b>Temporal</b>	<b>0.444</b>	<b>0.648</b>	<b>0.750</b>
	Aggregated	1.000	1.000	0.667
	Average	0.444	0.444	0.222

sion in a traditional society where a few merchants travel but most people stay in their villages. A TMG  $G(\eta, v, \gamma, p, b, d)$  is defined by six parameters, the number  $\eta$  of merchants, the number  $v$  of villages, the number  $\gamma$  of residents in each village, and three other parameters  $p$ ,  $b$ , and  $d$  to control the interconnections between residents in a village. Start with  $v$  mutually exclusive random graphs  $G_{\gamma, p}$  and  $\eta$  merchant nodes which are only connected to a resident node in a random graph; at every time step, the existing edges are removed and/or the new edges are added as follows:

(a) *Internal movement*: For each village  $G^i = (V^i, E^i)$  where  $i \in \{1, \dots, v\}$ , an existing edge  $e \in E^i$  will disappear with probability  $d$ ; while a nonexisting edge  $\hat{e} \notin E^i$  will appear with probability  $b$ .

(b) *External movement*: For each merchant node  $v^j$  where  $j \in \{1, \dots, \eta\}$ , an existing edge  $(v^j, v^{\text{old}})$  will disappear and then a new edge  $(v^j, v^{\text{new}})$  will appear with the mobility probability  $\text{Prob}_{\text{mobility}}(v^j)$ , where  $v^{\text{old}}$  and  $v^{\text{new}}$  are resident nodes in villages  $G^{\text{old}}$  and  $G^{\text{new}} (\neq G^{\text{new}})$ , respectively.

We use the probability  $\text{Prob}_{\text{mobility}}(\cdot)$  to differentiate the mobility of each merchant. In other words, merchant  $u$  moves with the probability  $\text{Prob}_{\text{mobility}}(u)$ . We here set  $\text{Prob}_{\text{mobility}}(u) \geq 0.5$ . Each merchant moves with a probability randomly assigned between 0.5 and 1.

Only our temporal metrics capture the merchants' mobility in TMG effectively. To show this, we generate 100 traveling merchant graphs with  $\eta = 1$ ,  $v = 5$ ,  $\gamma = 6$ ,  $p = 0.4$ ,  $b = 0.1$ , and  $d = 0.1$  during 100 steps, and compute the mean values of the temporal degree, closeness, and betweenness centrality metrics, respectively, for merchant and resident nodes. For comparison, we also compute the mean values of the centrality metrics in the aggregated graph and the average centrality metrics in the time series of graphs, respectively. Figure 4 demonstrates these simulation results.

If we use each node's average centrality values to measure its relative importance as it is computed from snapshots, the many temporal paths through the merchant nodes are ignored. Even though the centrality values in the aggregated graph can distinguish merchant node from resident nodes well, these centrality values are highly overestimated because the interaction frequency is ignored. To show this, we analyze the relation between the merchants' centrality values and

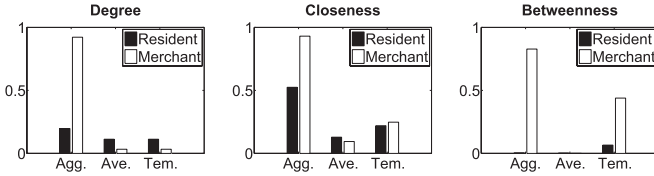


FIG. 4. The comparison between the centrality—degree (left), closeness (middle), and betweenness (right)—of the resident and merchant nodes in the aggregated graph, the average centrality of those in the time series of graphs, and the temporal centrality of those for TMGs with  $\eta = 1$ ,  $\nu = 5$ ,  $\gamma = 6$ ,  $p = 0.4$ ,  $b = 0.1$ , and  $d = 0.1$ .

mobility by calculating the Pearson correlation coefficients among them. The results are shown in Table III.

There are some missing values here. In the aggregated graphs, a merchant node is always on all shortest paths between all pairs of resident nodes belonging to different villages it has interacted with, regardless of its mobility, so its betweenness value is always fixed; and as the Pearson correlation is defined only if both standard deviations are finite and nonzero, we cannot define Pearson correlation coefficients between merchant nodes’ mobility and their betweenness. The same applies to temporal degree, average degree, and average betweenness (the standard deviation of a merchant node’s degree is zero since it is always connected to only one resident node, and its betweenness is zero in each snapshot since it is always an end node of a path). So it is hard to identify the high-mobility nodes such as the merchants in the TMG from their betweenness values in an aggregated graph; we do not even recommend using aggregated degree and closeness metrics.

On the other hand, the merchant nodes’ temporal closeness and betweenness values are highly correlated with their mobility. This is natural; a node with high mobility has high centrality in dynamic networks since the merchant nodes’ other conditions are all identical. We found that the merchant node with the highest mobility can be identified with a high probability in a TMG with multiple merchants by computing their temporal closeness or betweenness values. We generated 100 TMGs with  $\eta = 4$ ,  $\nu = 5$ ,  $\gamma = 6$ ,  $p = 0.4$ ,  $b = 0.1$ , and  $d = 0.1$ . We tried to identify the merchant node with the highest mobility by selecting the merchant node with the highest centrality value for each metric. When the merchant node with the highest centrality value is not unique, we arbitrarily choose one of the nodes with the same highest centrality value. Table IV shows the accuracy of such selection based on each centrality metric. In this table, we can see that the high-mobility nodes are identified with high probability using the temporal closeness or betweenness.

TABLE III. The correlation analysis between merchants’ centrality values and mobility for each metric type.

Type	Degree	Closeness	Betweenness
<b>Temporal</b>		<b>0.936</b>	<b>0.883</b>
Aggregated	0.645	0.653	
Average		0.088	

TABLE IV. The detection accuracy for the highest-mobility merchant node.

Type	Degree	Closeness	Betweenness
<b>Temporal</b>	<b>20%</b>	<b>72%</b>	<b>80%</b>
Aggregated	47%	47%	50%
Average	20%	19%	20%

## V. EFFECTIVENESS OF TEMPORAL CENTRALITY ON REAL DYNAMIC NETWORKS

Our next test for our proposed metrics is to try them out on real-world data sets and see whether they are really meaningful in practice. When the network topology can change over time as new edges are created and existing ones removed, can we use temporal centrality, computed from the nodes’ contact history, to estimate the importance of nodes in the future? Of course, if future human behavior were independent of past contact patterns, this task would be pointless; but we know already that societies do not work like that. In this section we discuss the implication of temporal centrality metrics in the real world. For brevity we will consider only temporal closeness and betweenness.

We use two contact traces of real mobile devices carried by humans: the Bluetooth trace of students at the University of Cambridge Computer Laboratory in the Haggie project [12] and a similar Bluetooth trace of students and staff at MIT [13]. We shall refer to these as CAMBRIDGE and MIT, respectively. For MIT, we use the contact trace from the first 5 days of the fall semester [14] to compare the results with those for CAMBRIDGE—we note that the only 85 nodes rather than the full 100 nodes appeared during this period. Table V describes some characteristics of each data set. These data sets were constructed from mobile device collocation where participants were given Bluetooth-enabled mobile devices to carry around; when two devices came within Bluetooth range, the devices logged the collocation event.

We evaluate the effectiveness of the proposed centrality metrics by computing the “message propagation delay” between nodes on these data sets—we use the first half of each contact trace for training input (i.e., a known historical human contact trace) and the rest for testing (i.e., the future “unknown” human contact traces). Formally, given a trace on a time interval  $[0, T]$ , we use the terms *training trace* to indicate the first half of the human contact trace on the time interval  $[0, \lfloor T/2 \rfloor]$  and *testing trace* to indicate the human contact trace on the time interval  $[\tau, T]$  where  $\lfloor T/2 \rfloor < \tau < T$ . We compute the temporal centrality values of nodes in the dynamic graph  $G_{0, \lfloor T/2 \rfloor}^D$  generated from the training trace and then test whether

TABLE V. Experimental data sets.

	CAMBRIDGE	MIT
Number of nodes	12	100
Start date	25 Jan 2005	26 Jul 2004
Duration	5 days	280 days
Av. contacts per day	846	231
Scanning rate	2 min	5 min

these values are really meaningful with a testing trace on the time interval  $[\tau, T]$ .

To evaluate the test performance, given a testing trace on the time interval  $[\tau, T]$ , we define the following metrics:

(a)  $\lambda(v, u)$ : The message propagation delay from node  $v$  to node  $u$  in the testing trace. The message propagation delay from  $v$  to  $u$  specifies how much time has elapsed from the time  $\tau$  to the time when  $v$  first meets  $u$ . If there is no contact between  $v$  and  $u$ ,  $\lambda(v, u)$  is defined as  $\infty$ . We note that  $\lambda(v, u)$  on the time interval  $[\tau, T]$  is different from the temporal shortest path distance  $\Delta_{\tau, T}(v, u)$ ;  $\lambda(v, u)$  can be computed in a totally independent way, regardless of the underlying dynamic graph. We use  $\lambda(v, u)$  rather than  $\Delta_{\tau, T}(v, u)$  to discuss the effects of the time window size  $w$  later since  $\Delta_{\tau, T}(v, u)$  is generally changed with  $w$ .

(b)  $\lambda_{\text{from}}(v)$ : The average message propagation delay from node  $v$  to all the other nodes in the testing trace. This can be computed as follows:

$$\lambda_{\text{from}}(v) = \frac{1}{|V| - 1} \sum_{u \in V \setminus v} \frac{1}{\lambda(v, u)}.$$

Here we assume  $1/\infty = 0$ . This metric is used to quantify in practical terms how quickly the node  $u$  can communicate with all other nodes at time  $\tau$ . We test whether  $\lambda_{\text{from}}(v)$  computed from a testing trace increases with  $v$ 's temporal closeness computed from the training trace.

(c)  $\lambda_{\text{sans}}(v)$ : The average message propagation delay between all nodes in the testing trace except the contacts related to node  $v$ . This can be computed as follows:

$$\lambda_{\text{sans}}(v) = \frac{1}{(|V| - 1)(|V| - 2)} \sum_{\substack{u, w \in V \setminus v \\ u \neq w}} \frac{1}{\lambda(u, w)}.$$

This metric is used to quantify how much communication speeds between all pairs of nodes are affected by removing the node  $u$ . We test whether  $\lambda_{\text{sans}}(v)$  computed from the testing trace decreases with  $v$ 's temporal betweenness centrality computed from the training trace.

We first compute the temporal closeness and betweenness centrality of nodes in the dynamic graph  $G_{0, [T/2]}^D$  generated from the training trace of each data set where  $w$  is set to the finest window granularity, corresponding to the device scanning rate (for example, 120 s for CAMBRIDGE) and plot them in Fig. 5. The computed values are sorted in descending order. This figure clearly shows that there is a small number of nodes with extremely high temporal centrality, and a large number of nodes with moderate or low centrality values, across all experiments except for the closeness in CAMBRIDGE. This implies that there exist nodes with high temporal centrality in practice.

We then analyze the correlation between  $\lambda_{\text{from}}$  (or  $\lambda_{\text{sans}}$ ) and closeness (or betweenness) centrality over nodes to discover how much useful centrality information is provided. We find that greater temporal centrality in the training trace is positively related to  $\lambda_{\text{from}}$  or  $\lambda_{\text{sans}}$  in the testing trace. We calculate the Kendall's tau (rank) correlation coefficients [15] between the  $\lambda_{\text{from}}$  (or the reverse of the  $\lambda_{\text{sans}}$ ) ranking of nodes in the testing trace on the time interval  $[\tau, T]$ , and the closeness (or betweenness) centrality ranking in the dynamic graph

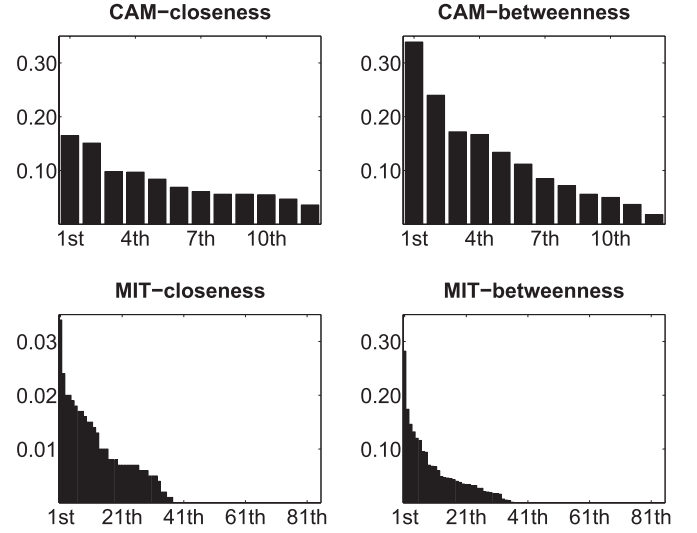


FIG. 5. Temporal centrality distribution of nodes: closeness distribution in CAMBRIDGE (top left), betweenness distribution in CAMBRIDGE (top right), closeness distribution in MIT (bottom left), and betweenness distribution in MIT (bottom right). The computed values are sorted in descending order. For improved visualization, we use the same range on the y axis except for the closeness centrality distribution in MIT (bottom left), since the centrality values in MIT are totally different from those in the other cases.

$G_{0, [T/2]}^D$  generated from the training trace. The correlation results obtained by varying  $\tau$  every 1 h from  $w(\lfloor T/2 \rfloor + 1)$  are shown in Fig. 6. For comparison, we also plot the correlation coefficients between  $\lambda_{\text{from}}$  (or  $\lambda_{\text{sans}}$ ) and the closeness (or betweenness) centrality in the aggregated and the average centrality metrics discussed above.

The temporal centrality metrics (red triangles) are clearly more effective than the aggregated (blue crosses) and average

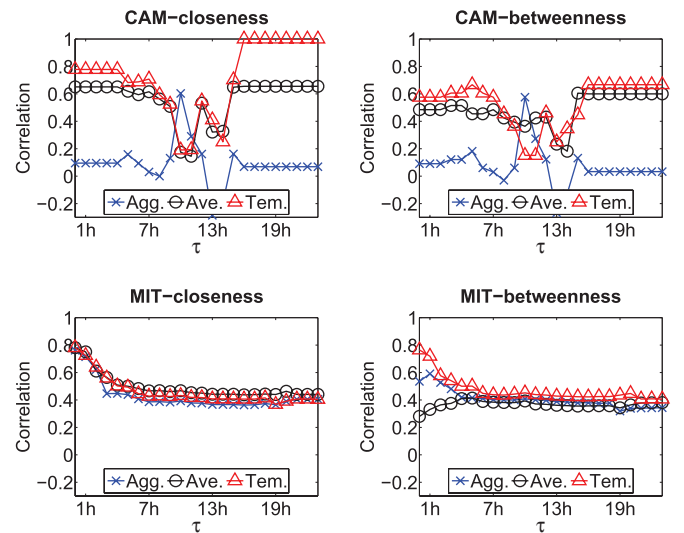


FIG. 6. (Color online) Kendall's tau (rank) correlation coefficients between  $\lambda_{\text{from}}$  (or  $\lambda_{\text{sans}}$ ) and closeness (or betweenness) centrality: closeness in CAMBRIDGE (top left), betweenness in CAMBRIDGE (top right), closeness in MIT (bottom left), and betweenness in MIT (bottom right).

(black circles) centrality metrics except for the closeness centrality in MIT. We note that all of the closeness metrics produced similar centrality ranking of nodes for the closeness centrality in MIT.

Interestingly, for CAMBRIDGE, we found fluctuation patterns when  $\tau$  is around  $w(\lfloor T/2 \rfloor + 1) + 13$  h while we cannot see such patterns for MIT. We surmise that underlying real-world differences may explain this. The contact trace at Cambridge was generated by 12 students with diverse lifestyles—some students are active at night, and others during the daytime. The contact trace at MIT came from a larger sample of 85 students and staff. Although a few students have diverse lifestyles, the centrality ranking of most MIT people is quite stable. Therefore the effects of external calendar cycles may be averaged over large samples. In other words, the Cambridge students' network centrality values change more dynamically over time than those of MIT students and staff. Also, for MIT, the effectiveness of the estimated centrality decreases with  $\tau$ . These results are natural consequences of our method since the relative importance of the training trace decreases when  $\tau$  increases. However, for CAMBRIDGE, this trend appears to be rather weak due to the periodic patterns of students over time.

Finally, we discuss the effect of varying the time window size  $w$ . As  $w$  increases, the temporal characteristics of human contacts are generally underestimated when time ordering and frequency of contacts within a time window are ignored, but the cost of computing centrality values decreases. In particular, the time complexity of the temporal closeness and betweenness centrality computation in a dynamic network  $G_{i,j}^D = (V, E_{i,j})$  can be dramatically cut since  $m$  generally dominates the overall time complexity of the centrality computation where  $m = j - i$ . By increasing  $w$  sufficiently, we can cut the total running time of both the temporal closeness and betweenness

computations because all nodes in  $V$  are  $O(|V|^2)$  and  $O(|V|^3)$ , respectively. Figure 7 shows the effects of varying  $w$  from 2 min for CAMBRIDGE (or 5 min for MIT) to 24 h. To demonstrate this we fix  $\tau = w(\lfloor T/2 \rfloor + 1)$ .

Unlike our expectation, the Kendall's tau (rank) correlation coefficients between the  $\lambda_{\text{from}}$  (or the reverse of the  $\lambda_{\text{sans}}$ ) ranking of nodes and the closeness (or betweenness) centrality ranking are almost stable with  $w$  although the correlation coefficients decrease slightly with  $w$  for the temporal closeness centrality ranking for CAMBRIDGE and the temporal betweenness centrality ranking for MIT. Thus a certain rough approximation of node centrality may often be almost as good as a fine-grained dynamic network. Intuitively, if students' contact patterns are determined by hourly class changes or half-daily moves between departments and colleagues, this level of granularity may be enough for most modeling purposes. Moreover, the average centrality metrics produce results similar to those of the temporal centrality metrics as  $w$  increases. Such insights as these enable us to tune our techniques to improve centrality prediction accuracy for a given budget.

## VI. CONCLUSION

We proposed temporal centrality metrics based on a simple but powerful model, the time-ordered graph, which embeds a dynamic network in a larger static one with directed flows. Our centrality metrics provide a simple extension in principle of the existing static metrics to the dynamic case.

We demonstrated their robustness and usefulness by applying them first to a number of interesting edge cases, such as where paths extend across a number of temporal snapshots, where a clique in a single snapshot obscures dynamic behavior at other times, and where connectivity depends on a small number of highly mobile vertices or edges. These cases show the inadequacy of existing attempts to extend static metrics to the dynamic case, and our simulation results show that our definitions are both feasible and robust.

Finally we applied our metrics to data sets from two real-world human contact networks. Our analysis showed a clear difference between the two networks; the Cambridge students' network centrality values change more dynamically over time than those of the MIT students and staffs. This may reflect the underlying real-world fact that the 12 students at Cambridge may have different life cycles while the effects of external calendar cycles may be averaged over large samples at MIT. That is an encouraging initial indicator. We offer our metrics to the research community as a better tool to measure behavior in dynamic networks. As part of this ongoing study, we plan to analyze the reciprocal relationship between temporal centrality and external (calendar) time to study how to choose training samples to improve the effectiveness of temporal centrality.

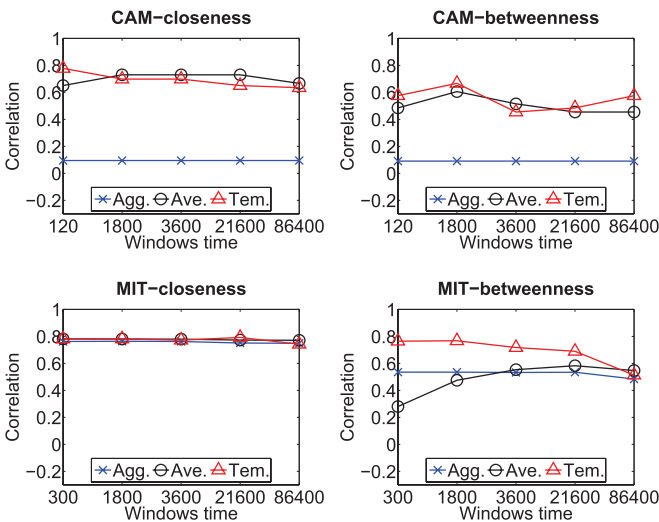


FIG. 7. (Color online) Kendall's tau (rank) correlation coefficients between  $\lambda_{\text{from}}$  (or  $\lambda_{\text{sans}}$ ) and the closeness (or betweenness) centrality by varying  $w$  while fixing  $\tau = w(\lfloor T/2 \rfloor + 1)$ : closeness in CAMBRIDGE (top left), betweenness in CAMBRIDGE (top right), closeness in MIT (bottom left), and betweenness in MIT (bottom right).

## ACKNOWLEDGMENT

The research of H. Kim was funded by Northrop Grumman Systems Corporation.



- [1] *The Structure and Dynamics of Networks*, edited by M. E. J. Newman, A. L. Barabási, and D. J. Watts (Princeton University Press, Princeton, 2010).
- [2] R. Albert, H. Jeong, and A. Barabasi, *Nature (London)* **406**, 378 (2000).
- [3] S. Nagaraja and R. Anderson, in *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, edited by Ross Anderson (Cambridge, UK, 2006).
- [4] H. Kim, J. Tang, R. Anderson, and C. Mascolo, *Comput. Networks* **56**, 983 (2012).
- [5] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia, in *Proceedings of the 3rd Workshop on Social Network Systems, SNS '10* (ACM, New York, 2010), pp. 3:1–3:6.
- [6] D. Kempe, J. Kleinberg, and A. Kumar, *J. Comput. Syst. Sci.* **64**, 820 (2002).
- [7] A. Ferreira, *Network, IEEE* **18**, 24 (2004).
- [8] V. Kostakos, *Physica A* **388**, 1007 (2009).
- [9] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, *SIGCOMM Comput. Commun. Rev.* **40**, 118 (2010).
- [10] J. Tang, C. Mascolo, M. Musolesi, and V. Latora, in *Proceedings of the 12th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2011)* (IEEE, Lucca, Italy, 2011), pp. 1–9.
- [11] T. Opsahl, F. Agneessens, and J. Skvoretz, *Social Networks* **32**, 245 (2010).
- [12] <http://www.haggleproject.org>.
- [13] N. Eagle and A. (Sandy) Pentland, *Personal Ubiquitous Comput.* **10**, 255 (2006).
- [14] <http://web.mit.edu/registrar/www/calendar0405.html>.
- [15] J. L. Myers, A. D. Well, and R. F. Lorch, *Research Design and Statistical Analysis* (Lawrence Erlbaum Associates, Hillsdale, NJ, 2003).