

# What if CLIQUE were fast? Maximum Cliques in Information Networks and Strong Components in Temporal Networks

Ryan A. Rossi, David F. Gleich, Assefaw H. Gebremedhin \*    Md. Mostofa Ali Patwary †

## Abstract

Exact maximum clique finders have progressed to the point where we can investigate cliques in million-node social and information networks, as well as find strongly connected components in temporal networks. We use one such finder to study a large collection of modern networks emanating from biological, social, and technological domains. We show inter-relationships between maximum cliques and several other common network properties, including network density, maximum core, and number of triangles. In temporal networks, we find that the largest temporal strong components have around 20-30% of the vertices of the entire network. These components represent groups of highly communicative individuals. In addition, we discuss and improve the performance and utility of the maximum clique finder itself.

**Keywords** max-clique, temporal strong components, social networks, information networks

## 1 Introduction

The MAX-CLIQUE problem is simply stated:

Consider a graph  $G$ . A clique of size  $k$  in  $G$  is a subset of  $k$  vertices that forms a complete subgraph. The maximum clique problem is to find the largest  $k$  such that there is a clique of size  $k$  in  $G$ .

Determining if a graph has a clique of size  $k$  is well known to be NP-complete. Approximating the largest clique is also known to be hard [18]. Clique is even hard for networks with power-law degree distributions and large clustering coefficients [1]. Clearly, MAX-CLIQUE is not, generally, fast. Yet, many real-world problems have features that do not elicit worst-cast behaviors from well-designed algorithms. In this manuscript, we use a state-of-the-art *exact maximum clique finder* [30] to investigate cliques in social and information networks.<sup>1</sup> We also use a relationship between strongly connected components in temporal networks and maximum cliques

in a reachability graph to find these temporal strong components [5, 25].

We have three goals with this manuscript.

The first is to learn about the structure of social and information networks by comparing maximum clique size to other common network properties. We use 38 networks for this study, obtained from various domains, that vary in size from a few hundred edges to 100 million edges. The results are presented in Section 2. We find interesting relationships between the average degree and maximum clique for network of Facebook friendships. Also, we study various upper-bounds on the maximum clique size for these networks. Given a connected graph  $G$ , it can be shown that the *clique number*  $\omega(G)$ —the size of the largest clique in  $G$ —obeys the following chain of bounds

$$\delta(G) \leq \omega(G) - 1 \leq K(G) \leq \Delta(G)$$

where  $K(G)$  is the largest  $k$  for which there is a  $k$ -core in  $G$  (a  $k$ -core in  $G$  is a maximal induced subgraph of  $G$  in which every node has at least  $k$  neighbors),  $\delta(G)$  is the minimum degree in  $G$ , and  $\Delta(G)$  is the maximum degree in  $G$ . Further, for a connected graph containing at least one triangle, we can also show that  $\omega(G) \leq \sqrt{2 \cdot T(G)}$ , where  $T(G)$  is the maximum number of triangles incident on any node in  $G$ . We find that, for most real-world networks,  $K(G)$  is typically within a factor of five of  $\omega(G)$  whereas the ratio between  $\sqrt{2 \cdot T(G)}$  and  $\omega(G)$  can be fairly large.

Inspired by [25, 5], our second goal is to compute the largest temporal strong components. When each edge represents a contact—a phone call, an email, or physical proximity—between two entities at a specific point in time, we have a particular type of evolving network structure [12]. A temporal path in such a network represents a sequence of contacts that obeys time. A temporal strong component is a set of vertices where all temporal paths exist. See Section 3 for a formal treatment. Just like CLIQUE, checking if an evolving network has a temporal strong component of size  $k$  is NP-complete. Yet, temporal strong components correspond to cliques in a temporal reachability graph where each edge signifies the existence of a temporal path between

\*Purdue University

†Northwestern University

<sup>1</sup>We use the terms *graph* and *network* interchangeably throughout the manuscript. Also, *vertex* and *node* are interchangeable too.

vertices. Thus, finding maximum cliques enables us to find temporal strong components. We investigate the size and structure of these cliques on phone call networks and Twitter retweet networks in Section 3. Perhaps surprisingly, the largest temporal strong components typically involve only a small fraction (20% to 30%) of the total vertices.

For a small portion of these experiments, where the networks are relatively dense, we find that the exact maximum clique method requires unacceptable time. In Section 4, we discuss extensions we made to the algorithm from [30] to reduce its runtime and increase its flexibility; in particular, we parallelize the clique-searches and use better upper-and-lower bounds on the maximum clique size to prune the search space further and allow additional flexibility.

We make all the software underlying this manuscript available in the spirit of reproducible research. See: <http://www.cs.purdue.edu/~dgleich/codes/cliques>

## 2 Maximum Cliques in Social and Information Networks

We begin by studying the properties of maximum cliques on modern networks. See Section 4 for a discussion of the maximum clique procedure we utilize in this section.

**2.1 Data.** For all of these experiments, we discard any edge weights, self-loops, and only consider the largest strongly connected component. In contrast to the temporal components we describe in the next section, in this section we mean the standard strong components. If the graph is directed, we then remove non-reciprocated edges. We present a complete summary of some basic network statistics for the following datasets in the supplementary material. In total, we consider 8 broad classes.

**1. Biological networks.** We study a network where the nodes are proteins and the edges represent protein-protein interactions (yeast [17]). We also study the celegans metabolic network [10] where the nodes are substrates and the edges are metabolic reactions. Cliques in these networks represent biologically relevant modules.

**2. Interaction networks.** Here, nodes represent individuals and edges represent interaction in the form of (i) email exchanges (email-enron [21], email-enron-employ[8]), (ii) message posts (wiki-Talk [19], fb-messages [27]), (iii) cellphone calls (reality [11]), or (iv) face-to-face contacts (infect-dublin, infect-hyper [16]).

**3. Social networks.** Nodes are again people, and edges represent social relationships in terms of (i) importance (wiki-Vote [19]), (ii) trust (epinions [31]), (iii) friendship or follower (orkut [24], youtube [23], soc-

LiveJournal [4], flickr [14], brightkite [7], gowalla [7], slashdot [21]).

**4. Facebook networks.** The nodes are people and edges represent “Facebook friendships” (fb-Stanford, fb-Penn94, fb-Berkeley, fb-Duke, fb-Texas84 [35]).

**5. Technological networks.** The nodes in these networks are either agents (p2p-Gnutella [22]) or routers (Internet-AS [15], as-Skitter [6], WHOIS [37], routers-rf[33]), and edges are observed communications between the entities.

**6. Collaboration networks.** These are networks in which nodes represent individuals and edges represent scientific collaborations (MathSciNet [28], ca-CondMat, ca-AstroPh, ca-HepPh [20]). Large cliques in these networks are expected because they are formed when papers have many authors. Thus, cliques are only interesting if they are larger than the author list of any particular paper.

**7. Web link networks.** Here, nodes are web-pages and edges are hyperlinks between pages (polblogs [2], wikipedia [13], web-Google [21]). The largest clique represents the largest set of pages where the author wishes users to freely move around.

**8. Retweet networks.** Finally, in retweet networks, nodes are Twitter users and edges represent whether the users have retweeted each other (political-retweets [9], twitter-copen [3]). We generated twitter-crawl ourselves, see the temporal network datasets for more information. Cliques are groups of users that have all mutually retweeted each other, and may represent an interest cartel.

**2.2 Analysis.** The time taken for the exact clique finder, a clique heuristic, the size of the maximum clique, and the size of the heuristic clique are shown in Table 1. The identified cliques range in size from 3 nodes to 239 nodes. In all but one case the clique finder from [30] identified the largest clique. Even when it failed to find the largest clique on the flickr network, we have an upper-bound from the max  $k$ -core that the clique is no larger than 309 vertices. In general, the method worked quickly, however, on the 106M edge orkut social network, it took almost 10 hours to find it.

There are some interesting surprises in these clique numbers. First, the technological networks have large cliques with 50 and 60 vertices. We did not expect to find such large cliques in these networks as cliques ought to represent redundant edges that could be more efficiently handled by alternative topologies. Note that this finding may represent a by-product of the data collection methodology, rather than a property of the network. Second, the largest clique in a social network or Facebook network has 51 members. These are fairly

Table 1: Max Cliques in Real-world Networks.

Graph	Max Clique		Time (seconds)			
	V	E	Exact	Heur	Exact	Heur
YEAST	1458	1948	6	6	0.04	0.05
CELEGANS	453	2025	9	8	0.04	0.04
MATHSCINET	332K	820K	25	25	0.05	0.07
CA-CONDMAT	21K	91K	26	26	0.03	0.02
CA-ASTROPH	17K	196K	57	57	0.08	0.04
CA-HEPPH	11K	117K	239	239	0.11	0.21
FB-MESSAGES	1266	6451	5	5	0.03	0.01
REALITY	6809	7680	5	5	0.05	0.04
ENRON-ONLY	143	623	8	8	0.06	0.04
INFECT-HYP	113	2196	15	13	0.03	0.04
WIKI-TALK	92K	360K	15	12	1.52	0.06
INFECT-DUB	410	2765	16	16	0.04	0.04
ENRON-LARGE	33K	180K	20	18	0.41	0.06
RETWEET-POL	96	117	4	3	0.03	0.02
TWITTER-COP	761	1029	4	4	0.06	0.06
RETWEET-CRA	1.1M	2.2M	13	13	0.74	0.22
WIKI-VOTE	889	2914	7	6	0.03	0.06
EPINIONS	26K	100K	16	12	0.08	0.04
YOUTUBE	495K	1.9M	16	13	3.12	0.42
SLASHDOT	70K	358K	26	23	8.54	0.07
BRIGHTKITE	56K	212K	37	33	5.48	0.06
FLICKR	513K	3.1M	-	35	-	0.83
GOWALLA	196K	950K	29	29	0.68	0.13
ORKUT	2.9M	106M	47	43	32238.8	45.34
LIVEJOURNAL	4M	27.9M	214	214	2.47	2.47
DUKE	9885	506K	34	29	153.81	0.06
BERKELEY	22K	852K	42	36	32.19	0.17
PENN	41K	1.3M	44	42	21.25	0.22
STANFORD	11K	568K	51	46	655.57	0.11
TEXAS	36K	1.5M	51	43	3543.86	0.24
P2P-GNUTEL	62K	147K	4	4	0.04	0.06
INTERNET-AS	40K	85K	16	11	0.16	0.02
ROUTERS-RF	2113	6632	16	15	0.07	0.05
WHOSIS	7476	56K	58	51	9708.89	0.05
AS-SKITTER	1.6M	11M	67	61	1784.65	2.33
POLBLOGS	643	2280	9	9	0.04	0.04
WEB-GOOGLE	1299	2773	18	18	0.01	0.06
WIKIPEDIA	1.8M	4.5M	31	31	1.94	0.23

large groups for everyone to know each other. Third, and finally, the collaboration networks had large cliques, as we expect. We suspect that the clique with 239 members represents a single paper.

Now, we wish to study the relationships between these clique sizes and various graph properties. A full table of these results is given in the supplementary material. Here, we summarize with more informative figures. We show how the size of the clique varies with the global clustering coefficient (ratio of triangles to wedges), the average degree, and average triangles per node in Figure 1. In the clustering coefficient plot (a), we see little relationship between the clustering coefficient and the maximum clique, whereas we see a much strong relationship between the average number of triangles and clique size (c). In the average degree plot (b), we also see the set of Facebook networks and the orkut network form a small group. Among these figures, the strongest relationship with the clique size is the log of the average triangle count.

To conclude our analysis of these networks, we study

the bounds on the size of the largest clique from the introduction:

$$\omega(G) - 1 \leq K(G) \text{ and } \omega(G) - 1 \leq \sqrt{2T(G)}.$$

These bounds are shown in Figure 2. We find that the  $k$ -core bound is accurate up to a factor of 5, and usually much more accurate. In the supplementary material, we further include the fraction of the maximum clique contained with the highest  $k$ -core. This value is usually high, except for the largest networks. This finding suggests that as graphs increase in size, the relationship between the  $k$ -core and maximum clique changes.

### 3 Temporal Strong Components

In this section, we use the maximum clique method as a subroutine to compute temporal strong components [25]. Since this area is somewhat new, we review some of the basic definitions in order to motivate the relationship between cliques and temporal strong components.

**DEFINITION 3.1. (TEMPORAL NETWORK)** *Let  $V$  be a set of vertices, and  $E_T \subseteq V \times V \times \mathbb{R}^+$  be the set of temporal edges between vertices in  $V$ . Each edge  $(u, v, t)$  has a unique time  $t \in \mathbb{R}^+$ .*

For such a temporal network, a path represents a sequence of edges that must be traversed in increasing order of edge times. That is, if each edge represents a contact between two entities, then a path represents a feasible route for a piece of information.

**DEFINITION 3.2. (TEMPORAL PATHS)** *A temporal path from  $u$  to  $w$  in  $G = (V, E_T)$  is a sequence of edges  $e_1, \dots, e_k$  such that the  $e_1 = (u, v_1, t_1), \dots, e_k = (u_k, w, t_k)$  where  $v_j = u_{j+1}$  and  $t_j < t_{j+1}$  for all  $j = 1$  to  $k$ . We say that  $u$  is temporally connected to  $w$  if there exists a such temporal path.*

This definition echoes the standard definition of a path, but adds the additional constraint that paths must follow the directionality of time. Temporal paths are inherently asymmetric because of the directionality of time.

**DEFINITION 3.3. (TEMPORAL STRONG COMPONENT)** *Two vertices  $(u, w)$  are strongly connected if there exists a temporal path  $\mathcal{P}$  from  $u$  to  $w$  and from  $w$  to  $u$ . A temporal strongly connected component (tSCC) is defined as a maximal set of vertices  $C \subseteq V$  such that any pair of vertices in  $C$  are strongly connected.*

While the vertices of a strong component in a graph define an equivalency class, and hence, we can partition a network into components, the same fact is not true of temporal strong components. The vertices in a temporal

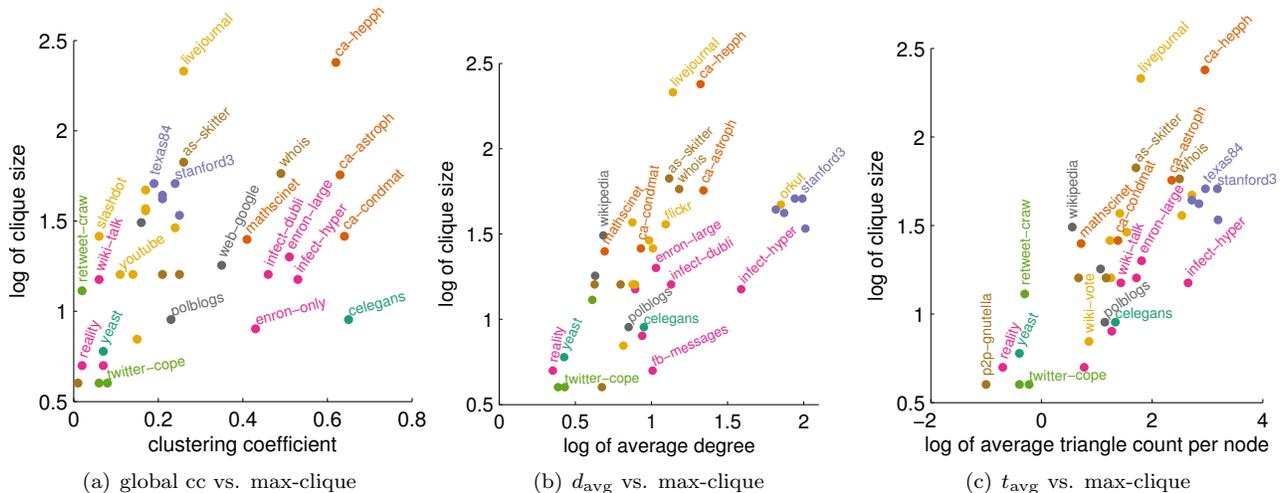


Figure 1: Each point shows one of the 38 graphs, colored by the graph type. See the discussion in the text.

strong component can overlap with those in another temporal strong component.

Note that a temporal weak component is always equal to the connected component in the static graph [34]. We conclude by mentioning that stronger definitions of temporal components exist. For example, the temporal paths used to define a strong component can be further restricted to only use vertices from the component  $C$  itself.

As previously mentioned, checking if a graph has a  $k$ -node temporal SCC is NP-complete [5, 25]. Nonetheless, we can compute the largest such strong component using a maximum clique algorithm. Let us briefly explain how.

The first step is to transform the temporal graph into what is called a strong-reachability graph. For each pair of vertices in  $V$ , we place an edge in the strong reachability graph if there is a temporal path between them. In Algorithm 1, this corresponds to the REACH subroutine. That method uses the temporal ordering proposed by [29] and builds up temporal paths backwards in time. With this reachability graph, the second step of the computation is to remove any non-reciprocated edges and then find a maximum clique. That maximum clique is the largest set of nodes where all pairwise temporal paths exist, and hence, is the largest temporal strong component [25].

**3.1 Temporal Graph Data.** We use three types of temporal graph data. For all networks, we discard self-loops and any edge weights. In all cases, the nodes represent people. And in all cases, the largest temporal strong components reflect groups of people that meet, interact, or retweet with each other sufficiently often to

---

**Algorithm 1** Largest Temporal Strong Component

---

**Input:** Dynamic Graph  $\mathcal{G} = (V, E_T)$

- 1 **procedure** MAX-TSCC( $\mathcal{G} = (V, E_T)$ )
  - 2  $E_R = \text{REACH}(\mathcal{G})$
  - 3 Remove non-reciprocal edges from  $E_R$
  - 4 Compute the MAX-CLIQUE in the graph  $(V, E_R)$
  - 5 Return the subgraph of  $G$  induced by  $C$
  
  - 6 **procedure** REACH( $\mathcal{G} = (V, E_T)$ )
  - 7 Sort edges to be in reverse time order
  - 8 Set  $E_R$  to be the set of all self-loops
  - 9 **for**  $(i, j, t) \in E$  **do**
  - 10     Add  $(i, k)$  to  $E_R$  for all  $k$  where  $(j, k) \in E_R$
  - 11 Return  $E_R$
- 

transmit any message or meme.

**Contact networks.** The edges are face-to-face contacts (infect-dublin, infect-hyper[16]). See ref. [32] for more details about these data.

**Interaction networks.** The edges are observed interactions such as forum posts (fb-forum [26]) and private messages (fb-messages [27]). We also investigate a cellular telephone call network where the edges are calls (reality [11]).

**Retweet networks.** Here, the edges are retweets. First, we investigate 26 retweet networks from various social and political hashtags. We collected these from Truthy [36] on September 20th, 2012. Of these, only a few had non-trivial temporal strong components which we discuss below. Also, we analyzed a network of political tweets centered around the November 2010

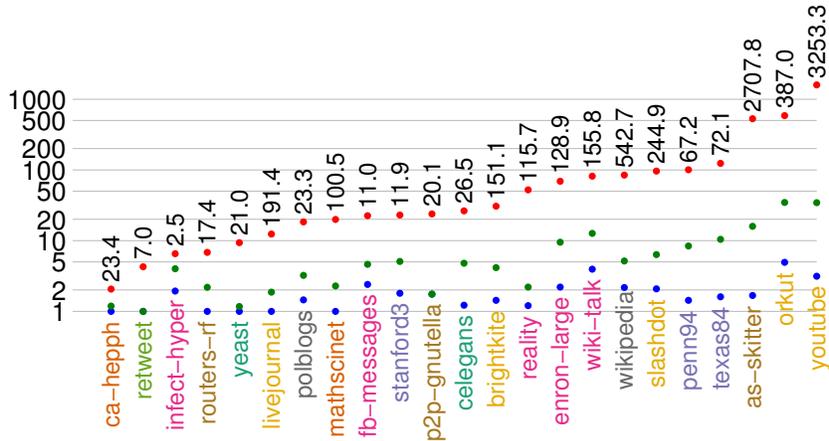


Figure 2: For a subset of 23 networks, sorted by increasing values of maximum degree over clique size, we show the relationship between the  $k$ -core upper bound on the clique size (shown as the blue points), the maximum triangle upper-bound on clique size (shown as the green points), and the maximum degree upper-bound on clique size (shown as the red points). We normalize these by the size of the maximum clique so all values are larger than one. Thus, we can immediately determine that the  $k$ -core bound is accurate up to a factor of 5, and usually much more accurate. The upper number is the ratio of maximum degree to average degree – a measure of the skewness of the degree distribution.

election in the US (retweet [9]). The final dataset is a retweet and mentions network from the UN conference held in Copenhagen. The data was collected over a two week period (twitter-copen [3]).

**3.2 Results and analysis** We now discuss the results of running our temporal strong component algorithms. Figures 3 and 4 show the reachability and largest temporal strong component from two networks, respectively. For the both networks, the exact clique finder did not complete within the specified period. We mark that fact with an “\*” to denote that it may not be the largest. The heuristic procedure found a clique with 1217 vertices for the reality network and 166 vertices for the retweets network. Given that the largest cores in these networks are 1236 and 175, respectively, in the worst case, we are only off by a few vertices. Our finding about the political party for the retweets also holds for the 166-core of the network, so we can be confident there is not a larger temporal strong component on the “political left.”

We summarize the remaining experiments on the temporal strong components in Tables 2 and 3. The first table presents basic network statistics of the original temporal graphs ( $G$ ) as well as their strong reachability graphs ( $\mathbf{R}_s$ ). For both cases, we list the largest clique we computed with either the exact or heuristic clique finder. Like above, those cliques marked with a “\*” represent heuristic computations because we only let the exact clique algorithm run for one hour. However, even for

these heuristics, we find that they are strikingly close to the upper-bound on the clique size given by the  $k$ -core. In fact, in all of the other examples, the largest clique in these networks *was* the largest  $k$ -core. Thus, just as in the political retweet network above, we are confident that our results are close to the true largest component.

The second table (Table 3) shows the time required for the maximum clique routine to return the largest temporal strong component – the largest clique in the reachability graph. As mentioned previously, we terminated the exact clique algorithm after an hour. In this table, we also include statistics on the largest strong component for the hashtag networks from Truthy. These show that there are heavily interacting groups of tweeters in these datasets that we can identify quickly (less than a second) using our methods. See the supplementary material for the network statistics on these hashtag networks.

Overall, we find a few interesting properties of these temporal strong components. In the contact networks (infect-hyper and infect-dublin), both of the largest strong components had about 100 vertices, despite the drastically different sizes of the initial dataset. We suspect this is a property of the data collection methodology because the infect-dublin data were collected over months, instead of days for the infect-hyper. In the interaction networks, the temporal components contain a significant fraction of the total vertices, roughly 20-30%. Finally, in the retweet

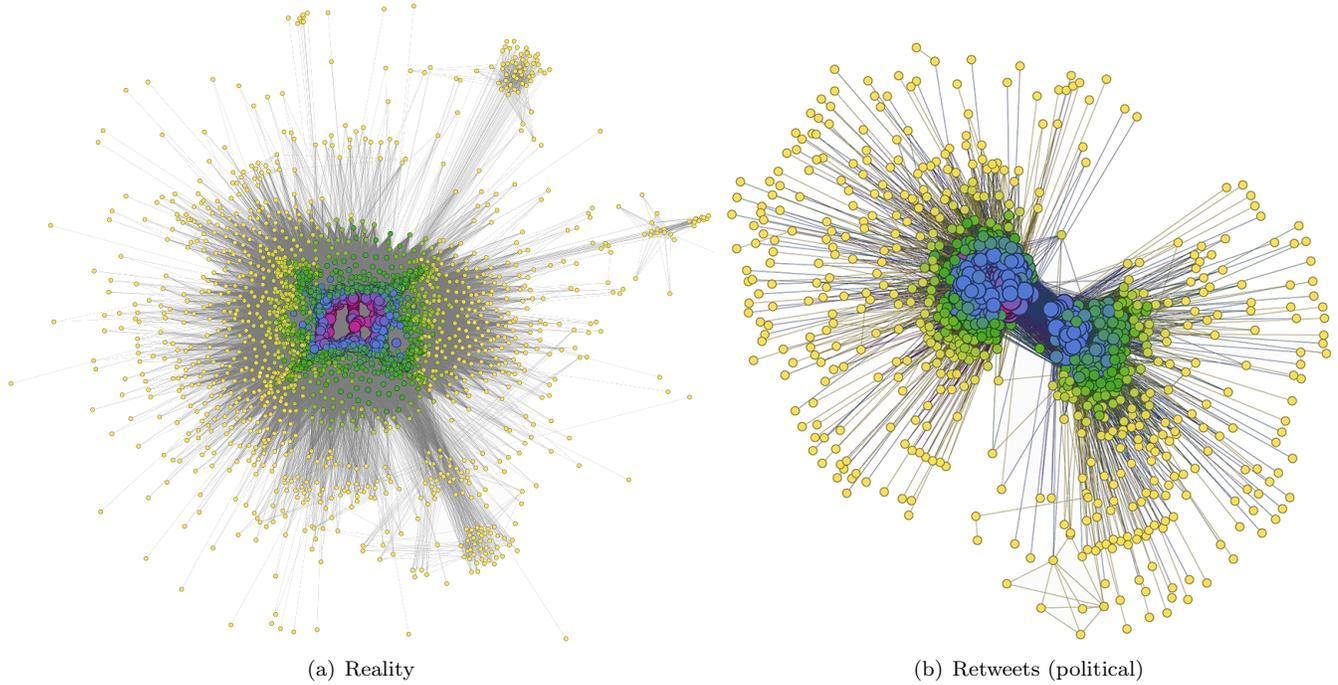


Figure 3: In order to compute the largest temporal strong component, we first compute the strong reachability network. These networks are rather dense and may reveal clear community structure as in the retweet networks (and, the infect-dublin network, see the supplementary material).

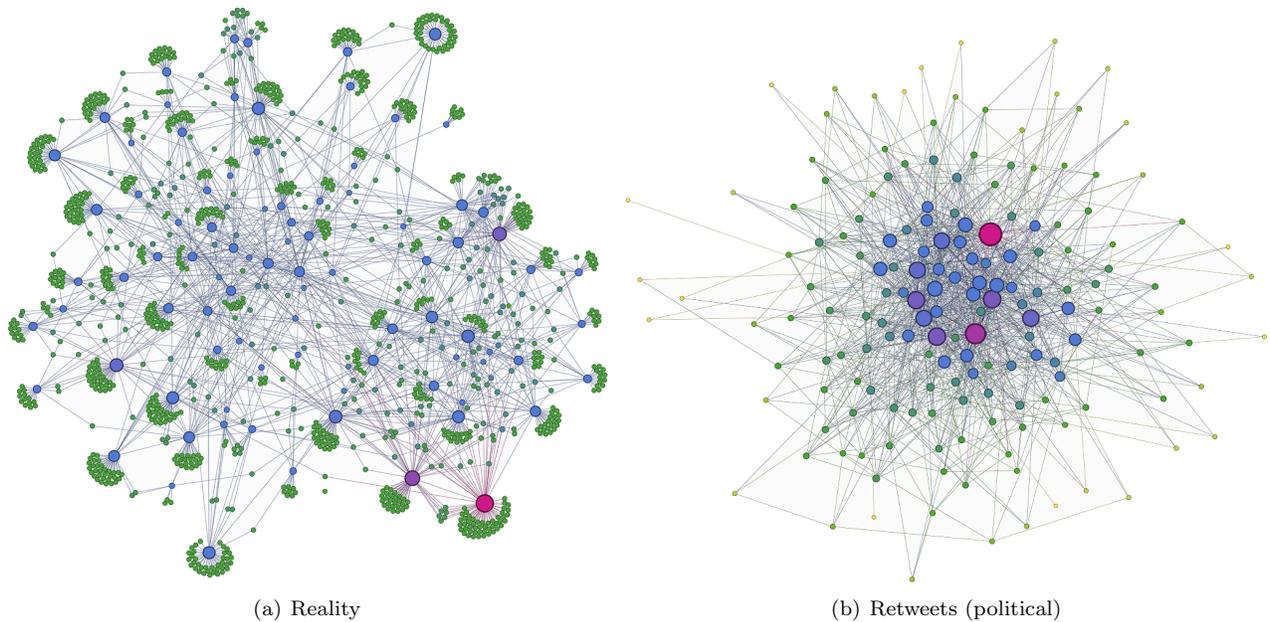


Figure 4: (a) In the largest\* (see discussion) temporal strong component of the reality network, we see a small group of core users maintaining connectivity among various groups, (b) whereas the largest\* component in the retweet network consists of 166 twitter users classified as politically “right” according to the original data with only a single exception. These users are a subset of one of the communities.

Table 3: For each of the temporal networks, we list the largest clique found by the exact method (if it finished in an hour) or via the heuristic, as well as the time taken by each algorithm. The exact method completes quickly for the contact networks, and hashtag networks, but has trouble with the interaction networks and retweet networks. In these cases, the heuristic gives a good lower-bound, that isn’t too far away from the  $k$ -core upper-bound.

Graph	$\omega$	$\tilde{\omega}$	Time (sec.)	
			Exact	Heur.
INFECT-DUBLIN	84	84	4.73	0.1
INFECT-HYPER	106	105	0.03	0.05
FB-FORUM	$\leq 273$	262	$> 3600$	1
FB-MESSAGES	$\leq 706$	677	$> 3600$	46.31
REALITY	$\leq 1236$	1217	$> 3600$	1039.55
RETWEET	$\leq 175$	166	$> 3600$	0.82
TWITTER-COPEN	$\leq 582$	573	$> 3600$	26.71
#ASSAD	8	7	$< 0.1$	$< 0.1$
#BAHRAIN	8	6	$< 0.1$	$< 0.1$
#SAUDI	8	8	$< 0.1$	$< 0.1$
#BARACKOBAMA	10	10	$< 0.1$	$< 0.1$
#ALWEFAQ	16	16	$< 0.1$	$< 0.1$
#JUSTINBIEBER	17	16	$< 0.1$	$< 0.1$
#OCCUPYWALLST	18	18	$< 0.1$	$< 0.1$
#GMANEWS	22	21	$< 0.1$	$< 0.1$
#ONEDIRECTION	27	27	$< 0.1$	$< 0.1$
#LOLGOP	42	40	0.1	$< 0.1$

networks, these temporal strong components are around 20% of the total vertices in the reachability graph. Given the strong communication pattern between these groups, they are good candidates for the center of a community in these networks.

## 4 Maximum Clique Algorithms

We now provide some algorithmic details of the maximum clique finders we have used in the previous two sections. We first outline the algorithms as introduced in [30] and then discuss the extensions we made in this work to improve performance and utility.

**4.1 The Exact Algorithm** Given an undirected graph  $G = (V, E)$ , let  $C_v$  denote a clique of the largest size containing the vertex  $v$ . A maximum clique in  $G$  can be found by computing  $C_v$  for every vertex  $v$  in  $V$  and then picking the largest among these. The exact algorithm we use follows this basic strategy, but drastically speeds up the process by avoiding the computation of every  $C_u$ ,  $u \in V$ , that would eventually be smaller than the global maximum. For such *pruning*

purposes, the algorithm maintains the size of the largest clique found so far ( $MaxSoFar$ ).

For a given vertex  $v$ ,  $C_v$  is evaluated by performing a *depth-first search* starting from  $v$ . At each *depth* of the search, the algorithm compares  $MaxSoFar$  against the number of remaining vertices that could potentially constitute a clique containing the vertex  $v$ . If the latter number is found to be smaller, the algorithm backtracks (the search is pruned). The algorithm employs several other pruning steps in addition to the backtracks. For example, a depth-first search from a vertex  $v$  is invoked only if the degree of  $v$  is greater than or equal to  $MaxSoFar$ , since otherwise the largest clique containing  $v$  would certainly be smaller than  $MaxSoFar$ .

**4.2 The Heuristic** In its search for a maximum clique in the entire graph, the exact algorithm described explores for every vertex  $v$  feasible cliques along a subset of the edges incident on  $v$ . The heuristic instead picks just *one* of these edges for further exploration. The picked edge is one incident on a vertex with the *largest degree* among the neighbors of  $v$ . The rationale for this choice is that the neighbor of  $v$  with the largest degree is more likely than other neighbors of  $v$  to be part of the largest clique on  $v$  ( $C_v$ ). The exact algorithm in the worst case has exponential runtime, whereas the runtime of the heuristic is bounded by  $O(|V| \cdot \Delta^2)$ , where  $\Delta$  is the maximum degree in the input graph.

**4.3 New Extensions** In this work, we extended both the exact algorithm and the heuristic in two major ways.

First, we *parametrized* the exact algorithm such that it takes both a lower bound  $lb$  and an upper bound  $ub$  on the size of the clique to be computed. The algorithm starts by setting  $MaxSoFar$  to  $lb$ —thus a good lower bound helps speed up the algorithm by allowing early pruning. And the algorithm stops the search as soon as a clique of size equal to  $ub$  is found—in the absence of a bound, the algorithm may continue to iterate over the remaining vertices even if the global maximum clique had already been found.

Second, we *parallelized* the algorithms. The computations of per-vertex maximum cliques ( $C_v$ ’s) over the vertices are independent of each other—and hence can be done in parallel—except for one issue: the need for information on the size of the global maximum clique found so far. However, since this information is needed only for pruning purposes, one need not insist on disseminating it to all processing units as soon as it is updated. Instead, one could work with a locally computed maximum clique for pruning purposes and occasionally exchange the global value at regular synchronization points. Following this strategy, we parallelized, using OpenMP

Table 2: Network statistics of the dynamic networks and their strong reachability graphs. The cases where the max  $k$ -core gives the exact size of the max clique are highlighted in gray. Here,  $d_{\max}$  and  $d_{\text{avg}}$  are the largest and average degrees, respectively. Also,  $\bar{\kappa}$  is the mean clustering coefficient;  $T$  and  $T_{\text{avg}}$  are the maximum number and average number of triangles incident on a vertex;  $K$  is the largest degree for a  $k$ -core to exist, and  $\omega$  is the clique number (or a lower-bound from the heuristic if the value has an \*).

Graph		$ V $	$ E $	$d_{\max}$	$d_{\text{avg}}$	$\bar{\kappa}$	$T$	$T_{\text{avg}}$	$\sqrt{2T}$	$K$	$\omega$
INFECT-HYPER	G	113	20818	98	38.9	0.53	1731	448	59	28	<b>15</b>
	$\mathbf{R}_s$	113	6222	112	110.1	0.99	6110	5934	111	105	<b>106</b>
INFECT-DUBLIN	G	10972	415912	64	8.1	0.45	315	22	25	18	<b>16</b>
	$\mathbf{R}_s$	10972	175573	219	32.0	0.80	9129	633	135	83	<b>84</b>
REALITY	G	6809	52050	261	2.3	0.02	52	0	10	5	<b>5</b>
	$\mathbf{R}_s$	6809	4714485	6069	1384.8	0.87	4695661	971909	3065	1235	<b>1217*</b>
RETWEET	G	18470	61157	785	2.6	0.01	238	1	22	5	<b>3</b>
	$\mathbf{R}_s$	1206	65990	551	109.4	0.70	52407	9073	324	174	<b>166</b>
TWITTER-COPEN	G	8581	45933	215	3.4	0.07	1210	3	49	5	<b>4</b>
	$\mathbf{R}_s$	2623	473614	1516	361.1	0.66	469039	111007	969	582	<b>573*</b>
FB-FORUM	G	899	33720	128	15.7	0.06	407	16	29	14	<b>3</b>
	$\mathbf{R}_s$	488	71011	435	291.0	0.86	70402	44845	375	272	<b>262*</b>
FB-MESSAGES	G	1899	61734	255	14.6	0.11	1095	23	47	20	<b>5</b>
	$\mathbf{R}_s$	1303	531893	1253	816.4	0.85	530538	317651	1030	706	<b>677*</b>

threading, both the exact algorithm and the heuristic.

The first improvement, parameterization, makes possible added flexibility. By intentionally setting  $ub$  to be smaller than a true upper bound on a maximum clique, a user may force the algorithm to terminate when a clique of the desired size ( $ub$ ) is found. Figure 5 illustrates the benefit of this. It shows how the runtime of the algorithm varies as the  $ub$  is varied for three large networks from our dataset (*as-skitter*, *texas*, and *whois*). Note that, in general, the algorithm detects cliques of small sizes extremely fast: in fact cliques of up to 90% of the maximum clique size can be computed at under 5% of the time it takes to compute the maximum clique. The plots also provide some insight into the different scenarios the three networks represent: in the *as-skitter* network, computing a clique of one size takes nearly the same time as computing a clique of just about any other size—meaning that cliques are easy to find—whereas in the other two networks it is increasingly difficult to detect larger cliques.

## 5 Discussion

In the title, we hypothetically asked “What if CLIQUE were fast?” We found that a state-of-the-art clique finder can identify maximum cliques in sparse networks with hundreds of millions of edges in a few seconds, but in the worst case, a few hours. These results show that cliques in social and information networks are not always difficult to find and hint at a deeper structural

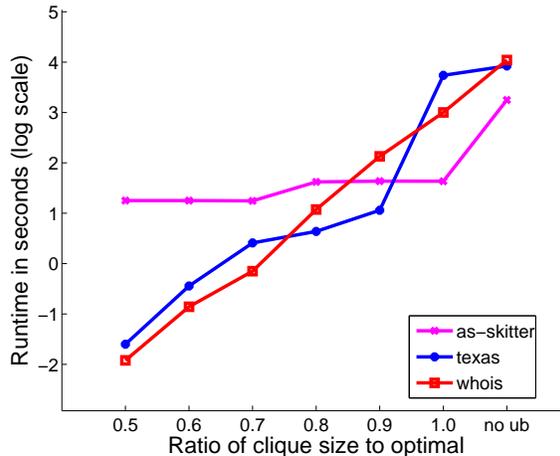


Figure 5: Runtime behavior as upper bound on the clique algorithm is varied.

property of these networks. The same maximum clique finder was less successful at identifying maximum cliques in temporal reachability networks (which give rise to temporal strong components). These networks have significant structure implied by the temporal edges that are hidden from the clique finder. This structure and the density of these networks may make these problems especially challenging. In the future, we plan to investigate additions to the clique finder to exploit such structure.

## References

- [1] J. Abello and M. Capalbo. Finding max-cliques in power-law graphs with large clique coefficients. Technical Report 2006-10, DIMACS, 2006.
- [2] L. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In *Link Discovery Workshop*, pages 36–43. ACM, 2005.
- [3] N. Ahmed, F. Berchmans, J. Neville, and R. Kompella. Time-based sampling of social network activity graphs. In *SIGKDD MLG*, pages 1–9, 2010.
- [4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *SIGKDD*, pages 44–54, 2006.
- [5] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. *ADHOC-NOW*, pages 259–270, 2003.
- [6] CAIDA. Skitter. <http://caida.org/tools/measurement/skitter/>.
- [7] E. Cho, S. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*, pages 1082–1090, 2011.
- [8] W. Cohen. Enron email dataset. <http://www.cs.cmu.edu/~enron/>. Accessed in 2009.
- [9] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, A. Flammini, and F. Menczer. Political polarization on twitter. In *ICWSM*, 2011.
- [10] J. Duch and A. Arenas. Community identification using extremal optimization phys. *Rev. E*, 72:027104, 2005.
- [11] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.
- [12] A. Ferreira. On models and algorithms for dynamic communication networks: The case for evolving graphs. In *ALGOTEL*, 2002.
- [13] D. Gleich, P. Constantine, A. Flaxman, and A. Gunawardana. Tracking the random surfer: empirically measured teleportation parameters in PageRank. In *WWW*, pages 381–390, 2010.
- [14] D. F. Gleich. Graph of flickr photo-sharing social network crawled in may 2006. In *DOI: 10.4231/D39P2W550*, 2012.
- [15] Internet-AS. Oregon RouteViews project. <http://www.routeviews.org/>.
- [16] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J. Pinton, and W. Van den Broeck. What’s in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011.
- [17] H. Jeong, S. Mason, A. Barabasi, and Z. Oltvai. Lethality and centrality in protein networks. *arXiv preprint cond-mat/0105306*, 2001.
- [18] S. Khot. Improved inapproximability results for max-clique, chromatic number and approximate graph coloring. In *FOCS*, pages 600–, 2001.
- [19] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.
- [20] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1):2, 2007.
- [21] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.
- [22] R. Matei, A. Iamnitchi, and P. Foster. Mapping the gnutella network. *Internet Computing*, 6(1):50–57, 2002.
- [23] A. Mislove. *Online Social Networks: Measurement, Analysis, and Applications to Distributed Information Systems*. PhD thesis, Rice University, CS, 2009.
- [24] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *SIGCOMM*, pages 29–42, 2007.
- [25] V. Nicosia, J. Tang, M. Musolesi, G. Russo, C. Mascolo, and V. Latora. Components in time-varying graphs. *Chaos*, 22(2):023101, 2012.
- [26] T. Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 2011.
- [27] T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.
- [28] G. Palla, I. Farkas, P. Pollner, I. Derényi, and T. Vicsek. Fundamental statistical features and self-similar properties of tagged networks. *New Journal of Physics*, 10(12):123026, 2008.
- [29] R. K. Pan and J. Saramäki. Path lengths, correlations, and centrality in temporal networks. *arXiv*, soc-ph:1101.5913v2, 2011.
- [30] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. keng Liao, and A. Choudhary. Fast algorithms for the maximum clique problem on massive sparse graphs. *arXiv preprint arXiv:1209.5818v2*, 2012.
- [31] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. *ISWC*, pages 351–368, 2003.
- [32] SocioPatterns. Infectious contact networks. <http://www.sociopatterns.org/datasets/>. Accessed 09/12/12.
- [33] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with rocketfuel. In *SIGCOMM*, volume 32, pages 133–145, 2002.
- [34] J. Tang, M. Musolesi, C. Mascolo, and V. Latora. Characterising temporal distance and reachability in mobile and online social networks. *SIGCOMM Computer Communication Review*, 40(1):118–124, 2010.
- [35] A. Traud, P. Mucha, and M. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 2011.
- [36] Truthy. Information diffusion research at indiana university. <http://truthy.indiana.edu/>. Accessed 10/20/12.
- [37] WHOIS. Internet routing registries. <http://www.irr.net/>.

## Appendix

Table 4: Network statistics of the static graphs from **section 2**, including the size of the maximum clique and corresponding upperbounds. Here,  $d_{\max}$  and  $d_{\text{avg}}$  are the largest and average degrees, respectively. Also,  $\bar{\kappa}$  is the mean clustering coefficient;  $T$  and  $T_{\text{avg}}$  are the maximum number and average number of triangles incident on a vertex;  $K$  is the largest degree for a  $k$ -core to exist,  $\omega$  is the clique number, and  $\Gamma_K = \frac{|C_n K|}{|C|}$  is the recall measurement by using the  $k$ -core to find the maximum clique.

Graph	$ V $	$ E $	$d_{\max}$	$d_{\text{avg}}$	$\bar{\kappa}$	$T$	$T_{\text{avg}}$	$\sqrt{2T}$	$K$	$\omega$	$\Gamma_K$
YEAST	1458	1948	56	2.7	0.07	18	0.4	6	5	<b>6</b>	1.00
CELEGANS	453	2025	237	8.9	0.65	870	21.7	42	10	<b>9</b>	0.89
MATHSCINET	332689	820644	496	4.9	0.41	1564	5.2	56	24	<b>25</b>	1.00
CA-CONDMAT	21363	91286	279	8.5	0.64	1615	24.0	57	25	<b>26</b>	1.00
CA-ASTROPH	17903	196972	504	22.0	0.63	11269	226.2	150	56	<b>57</b>	1.00
CA-HEPPH	11204	117619	491	21.0	0.62	39633	899.1	282	238	<b>239</b>	1.00
FB-MESSAGES	1266	6451	112	10.2	0.07	242	5.9	22	11	<b>5</b>	1.00
REALITY	6809	7680	261	2.3	0.02	52	0.2	10	5	<b>5</b>	0.80
ENRON-ONLY	143	623	42	8.7	0.43	125	18.7	16	9	<b>8</b>	1.00
INFECT-HYPER	113	2196	98	38.9	0.53	1731	447.8	59	28	<b>15</b>	0.93
WIKI-TALK	92117	360767	1220	7.8	0.06	17699	27.2	188	58	<b>15</b>	1.00
INFECT-DUBLI	410	2765	50	13.5	0.46	280	52.1	24	17	<b>16</b>	1.00
ENRON-LARGE	33696	180811	1383	10.7	0.51	17744	64.6	188	43	<b>20</b>	0.80
RETWEET	96	117	17	2.4	0.06	6	0.4	3	3	<b>4</b>	1.00
TWITTER-COPE	761	1029	37	2.7	0.08	27	0.6	7	4	<b>4</b>	1.00
RETWEET-CRAW	1112702	2278852	5070	4.1	0.02	1555	0.5	56	18	<b>13</b>	0.00
WIKI-VOTE	889	2914	102	6.6	0.15	251	7.2	22	9	<b>7</b>	0.57
EPINIONS	26588	100120	443	7.5	0.14	5151	18.0	101	32	<b>16</b>	0.00
YOUTUBE	495957	1936748	25409	7.8	0.11	151081	14.8	550	49	<b>16</b>	0.94
SLASHDOT	70068	358647	2507	10.2	0.06	13382	17.2	164	53	<b>26</b>	1.00
GOWALLA	196591	950327	14730	9.7	0.24	93817	34.7	433	51	<b>29</b>	0.00
FLICKR	513969	3190452	4369	12.4	0.17	524525	343.0	1024	309	<b>35*</b>	-
BRIGHTKITE	56739	212945	1134	7.5	0.17	11517	26.1	152	52	<b>37</b>	1.00
ORKUT	2997166	106349209	27466	71.0	0.17	1313133	525.1	1621	230	<b>47</b>	0.00
LIVEJOURNAL	4033137	27933062	2651	13.9	0.26	79740	62.1	399	213	<b>214</b>	1.00
DUKE14	9885	506437	1887	102.5	0.25	41982	1560.7	290	85	<b>34</b>	1.00
BERKELEY13	22900	852419	3434	74.4	0.21	69511	703.4	373	64	<b>42</b>	1.00
PENN94	41536	1362220	4410	65.6	0.21	68097	520.6	369	62	<b>44</b>	1.00
STANFORD3	11586	568309	1172	98.1	0.24	33177	1511.1	258	91	<b>51</b>	1.00
TEXAS84	36364	1590651	6312	87.5	0.19	141050	922.2	531	81	<b>51</b>	1.00
P2P-GNUTELLA	62561	147878	95	4.7	0.01	17	0.1	6	6	<b>4</b>	0.00
INTERNET-AS	40164	85123	3370	4.2	0.21	8513	4.7	130	23	<b>16</b>	1.00
ROUTERS-RF	2113	6632	109	6.3	0.25	588	14.8	34	15	<b>16</b>	1.00
WHOIS	7476	56943	1079	15.2	0.49	22271	314.0	211	88	<b>58</b>	1.00
AS-SKITTER	1694616	11094209	35455	13.1	0.26	564609	50.9	1063	111	<b>67</b>	0.00
POLBLOGS	643	2280	165	7.1	0.23	392	14.0	28	12	<b>9</b>	1.00
WEB-GOOGLE	1299	2773	59	4.3	0.35	189	11.7	19	17	<b>18</b>	1.00
WIKIPEDIA	1864433	4507315	2624	4.8	0.16	12404	3.6	158	66	<b>31</b>	0.00

Table 5: Network statistics of the **dynamic retweet graphs** and their strong reachability graphs from **section 3**.

<b>Graph</b>		$ V $	$ E $	$d_{\max}$	$d_{\text{avg}}$	$\bar{\kappa}$	$T$	$\sqrt{2T}$	$K$	$\omega$
#OCCUPYWAL	G	3609	3936	2811	1.1	0.02	8	3	1	<b>2</b>
	$\mathbf{R}_s$	127	931	112	14.7	0.72	809	28	17	<b>18</b>
#ALWEFAQ	G	4171	7123	865	1.7	0.07	77	9	4	<b>3</b>
	$\mathbf{R}_s$	72	355	39	9.9	0.50	280	17	15	<b>16</b>
#BAHRAIN	G	4676	8007	256	1.7	0.01	15	4	2	<b>3</b>
	$\mathbf{R}_s$	72	129	20	3.6	0.28	66	8	7	<b>8</b>
#MITTROMNE	G	7974	8597	1282	1.1	0.00	16	4	1	<b>2</b>
	$\mathbf{R}_s$	102	108	25	2.1	0.20	38	6	4	<b>5</b>
#GMANEWS	G	8373	8832	6146	1.1	0.00	98	10	1	<b>2</b>
	$\mathbf{R}_s$	135	1078	114	16.0	0.72	950	31	21	<b>22</b>
#JUSTINBIE	G	9405	9615	3176	1.0	0.00	2	1	1	<b>2</b>
	$\mathbf{R}_s$	62	442	42	14.3	0.69	375	19	16	<b>17</b>
#BARACKOBA	G	9631	9826	6107	1.0	0.00	5	2	1	<b>2</b>
	$\mathbf{R}_s$	80	226	43	5.7	0.48	153	12	9	<b>10</b>
#LOLGOP	G	9765	10324	8280	1.1	0.01	8	3	1	<b>2</b>
	$\mathbf{R}_s$	273	4510	249	33.0	0.73	4247	65	41	<b>42</b>

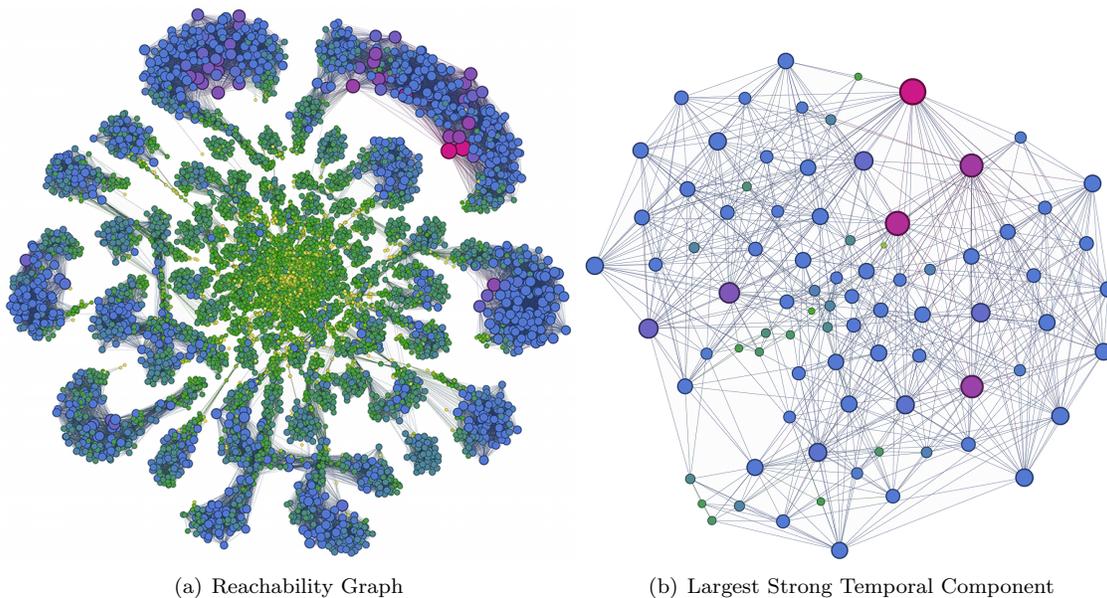


Figure 6: **Infectious Contact Network (Dublin) from Section 3**