Reachability Analysis and Modeling of Dynamic Event Networks

Kathy Macropol and Ambuj Singh

Department of Computer Science
University of California
Santa Barbara, CA 93106 USA
{kpm,ambuj}@cs.ucsb.edu

Abstract. A wealth of graph data, from email and telephone graphs to Twitter networks, falls into the category of dynamic "event" networks. Edges in these networks represent brief events, and their analysis leads to multiple interesting and important topics, such as the prediction of road traffic or modeling of communication flow. In this paper, we analyze a novel new dynamic event graph property, the "Dynamic Reachability Set" (DRS), which characterizes reachability within graphs across time. We discover that DRS histograms of multiple real world dynamic event networks follow novel distribution patterns. From these patterns, we introduce a new generative dynamic graph model, DRS-Gen. DRS-Gen captures the dynamic graph properties of connectivity and reachability, as well as generates time values for its edges. To the best of our knowledge, DRS-Gen is the first such model which produces exact time values on edges, allowing us to understand simultaneity across multiple information flows.

Keywords: Graph Generator, Dynamic Networks, Reachability.

1 Introduction

Vast amounts of graph datasets are generated each day by applications such as social networks, communication networks (like email graphs or Twitter), bioinformatics, and the Internet. The analysis and mining of these networks has been an active and important area of research, leading to both newly discovered fundamental network properties, as well as interesting and useful new knowledge and applications, from graph clustering for gene function discovery to network modeling for link or structure prediction [24,25,26].

Previous work on the analysis of graphs and their properties have analyzed degree distribution, number of triangles, relationships between the eigenvalues of the graph, etc [14,33,31]. These discovered properties have led to novel generative graph models, capable of producing new graph structures which capture and mimic such properties. Generative graph models have many interesting and important uses, including generation of synthetic datasets for analysis, graph anonymization, graph compression, prediction of graph and link evolution. While useful for many purposes, these graph models still mimic only the static network

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 442–457, 2012.
© Springer-Verlag Berlin Heidelberg 2012

graph structure. The addition of dynamic components to these static networks has been an active topic of research in the last few years, with much research concentrated on dynamic "state" networks (where links have a tendency to endure) such as in social or collaboration networks.

In contrast to dynamic state networks, links represent brief events in dynamic "event" networks. Examples of such graphs include communication networks like email or telephone call graphs. Dynamic event networks form a large and important category for graph datasets. However, their structure and dynamics do not fit well with many of the current dynamic network models. In dynamic event networks, timestamps associated with each link may convey the dynamics, allowing for both evenly spaced flows as well as arbitrarily long pauses or bursts of activity. This timed dynamic behavior is an integral part of a time-evolving network, and can be difficult to capture in models.

In this paper, we focus on dynamic event networks, looking especially at a new property, the "Dynamic Reachability Set" (DRS), which characterizes reachability within graphs across time. Reachability in general, along with concepts such as graph density or planarity, is a fundamental network property. In an evolving graph, reachability can convey latency of information flow between pairs of nodes in dynamically changing networks (e.g. mobile ad hoc or sensor networks); or latency along logistic/supply chain networks under dynamics; or even gossiping latency in dynamic social networks. Specifically, the DRS of a starting node consists of the set of all nodes reachable from the starting node, across a fixed time interval Δ. In this paper, we analyze the DRS properties, at a series of time intervals, for nodes within multiple real world dynamic event networks. From this analysis, we discover that DRS sizes follow a DGX distribution [6] for low values of Δ, but that this relation breaks as Δ increases. Additionally, we find that the rate in which the relation changes is specific to each network, but generally follow a log-normal-like curve.

The dynamic behaviors discovered from our analysis open the door for the learning and creation of new, novel generative modeling techniques which can now link time together with changes in network structure. Using this discovery, we focus on the generation of network dynamics, and propose a new generative modeling algorithm, DRS-Gen, able to produce dynamic graph structures that mimic the DRS properties of real world dynamic event networks across time. To the best of our knowledge, DRS-Gen is the first generative graph model able to assign and fit timestamps to edges, such that the rates of flow and reachability across a dynamic network are preserved. We introduce methods to learn the model parameters, and from there implement DRS-Gen, fitting this model to multiple real world dynamic event networks. From our results, we find that our generated graphs fit well and capture the flow and reachability distributions of real world graphs, across time, making it both a novel and potentially useful tool for generative graph modeling and analysis.

Overall, our main contributions come in three parts. First, we introduce and analyze a novel, relevant, and interesting new dynamic event graph property: the Dynamic Reachability Set. Second, we propose a new generative dynamic event

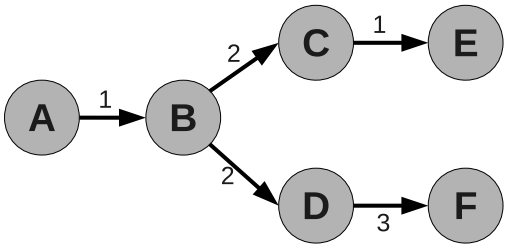

Fig. 1. An example dynamic event graph. The numbers on edges represent timestamps for the links. A time threshold of $\Delta = [1, 2)$ would give node A a DRS of {A,B}; for $\Delta = [1, 3)$, DRS = {A,B,C,D}; for $\Delta = [1, 4)$, DRS = {A,B,C,D,F}.

graph model, DRS-Gen, that allows for the generation of time-aware edges and flows. And third, we demonstrate not only how DRS-Gen may produce graphs, but also how it can be fit, naturally and easily, to real-world communication graphs, such that it may capture the reachability and dynamics of these graphs.

The rest of this paper is organized as follows. Section 2 introduces the concept of Dynamic Reachability Sets, as well as contains the analysis and results obtained from studying these sets on real world graphs. Section 3 introduces the DRS-Gen model, outlining the algorithm and theory behind it, then presenting and analyzing the results of our model when fit to multiple real world dynamic networks. Section 4 consists of a short survey of related work and previously introduced techniques on graph analysis and generation. Finally, in Section 5, we summarize our work, overviewing our contributions and conclusions.

2 Dynamic Reachability Sets

For this work, we focus our attention on the property of reachability within dynamic event networks. Specifically, let $G = \{V, E, T\}$ be a directed, dynamic graph where V are the set of vertices and $E = [e_1, e_2, \cdots, e_m]$ are the list of edges, where edge $e_i = (v_j, v_k)$ represents a link between nodes v_j and v_k. Additionally, the edges in E are ordered by the time function T, where $T(e_i)$ gives the timestamp for edge e_i. We can then define the DRS starting from node v_s and timestamp t_{start}, recursively over time interval Δ, as shown in Algorithm 1.

Algorithm 1. Calculate the DRS

Require: $\mathbf{DRS} := \{v_s\}, t_{start}, t_{end} := t_{start} + \Delta$
1: **procedure** CALCDRS($\mathbf{DRS}, t_{start}, t_{end}$)
2: **for all** v_i **where** $(e_k = (v_s, v_i), v_s \in \mathbf{DRS}, e_k \in E, t_{start} \leq T(e_k) < t_{end})$
3: $\mathbf{DRS} := \mathbf{DRS} \cup \{v_i\}$
4: **CalcDRS**($\mathbf{DRS}, T(e_k) + 1, t_{end}$)
5: **end procedure**

Figure 1 illustrates an example. In this graph, directed edges are associated with timestamps, and node A sends a message at timestamp 1 to node B. Node

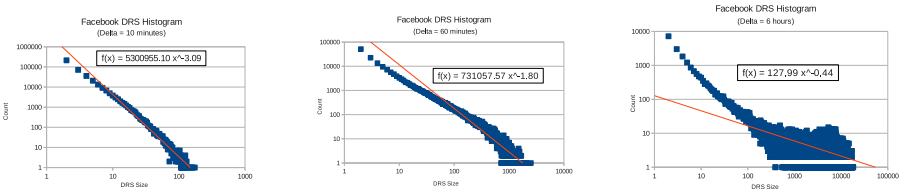

Fig. 2. Histograms of DRS size for the Facebook wall posts dataset, at increasing values for Δ. The decreasing slope of the approximately fit power law curve shows the movement of "mass" to the right as Δ increases.

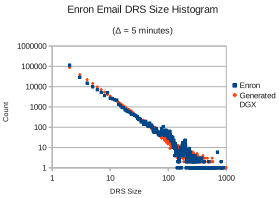

Fig. 3. Histogram of DRS size for the Enron Email dataset at $\Delta = 5$ minutes, along with the a generated DGX distribution ($\sigma = -7.67$, $\mu = 3.03$)

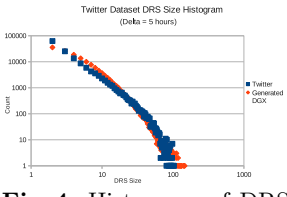

Fig. 4. Histogram of DRS size for the Twitter dataset at $\Delta = 5$ hours, along with the a generated DGX distribution ($\sigma = 1.10$, $\mu = 0.990$)

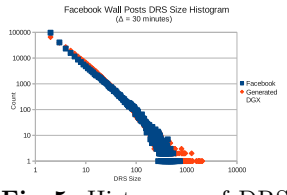

Fig. 5. Histogram of DRS size for the Facebook Wall post dataset at $\Delta = 30$ minutes, along with the a generated DGX distribution ($\sigma = 0.013$, $\mu = 1.88$)

B sends two messages, one to node C and one to node D at timestamp 2, etc. To find the DRS of node A, given the time interval Δ, we collect the set of all nodes that may be reached by recursively traveling edges within the time frame, starting from node A, without going backwards in time. This means that an interval starting at 1, with a Δ of 2, will include nodes A, B, C, D,and F, but not node E because to reach node C from node A takes until timestamp 2, and the edge from C to E occurred previously at timestamp 1.

The reachability set of a node, n_s, represents the nodes it is possible for n_s to reach across a specific time interval. Furthermore, the sequence of nodes and links followed to obtain the DRS can be thought of as a small "flow" within the graph. Overall, the set of DRS values for all nodes in a graph provides a window into the graph's dynamic connectivity and flow between all of its nodes. As the DRS time interval grows, we would expect the average DRS sizes to increase as well, since the number of links contained within the interval, and therefore the chances of reaching additional nodes, grows as well.

We collected and analyzed the actual DRS sets for multiple real world networks, and found that this intuition does indeed hold. Figure 2 shows the log-log plot of DRS size versus count, for a network consisting of Facebook wall posts and replies [34] containing 47K nodes (users) and 877K directed edges (wall posts from one user to the other). For each time interval, a series of non-overlapping time windows (for different values of t_1, the initial timestamp mentioned in Equation (1)) was used to discover DRS counts for each node in the graph. From the plots, we can see that as Δ increases from 10 minutes to 6 hours, more

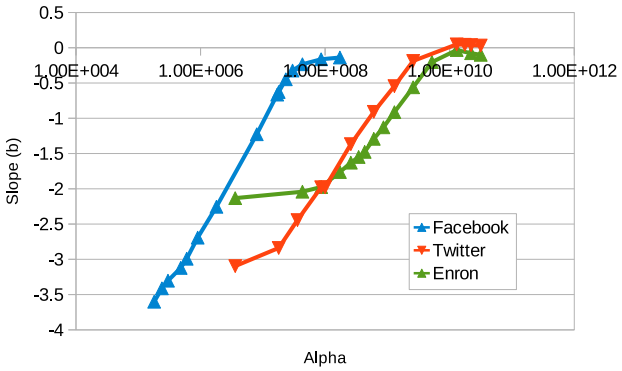

Fig. 6. Comparison of slope (for fit log-log curve) against Δ (logarithmic). It can be seen that various graphs each have their own rate of change. However, each follows a reversed log-normal form.

nodes gather on the right of the plot. Similar curves were found for other dynamic networks, as well (Figures 9 and 11 display additional histogram results for the Enron Email dataset [15] consisting of 87K unique email users and 360K directed email edges, as well as a crawled Twitter dataset [26] containing 8K Twitter users and 663K directed "reply to" and "retweet" edges). Again, this shift across time is largely reflected in the plots, and is echoed in the decreasing slope of the power-law curve loosely fit to the data. As the time interval increases, more nodes are able to reach larger amounts of the graph. Additionally, we can see that the shape of the curve undergoes an extreme change across time, as well. Additionally, it is apparent that the rate of change in the slope and curves varies, depending upon the graph. Figure 6 shows a plot of the slopes for the various networks, as they change across time, confirming that each network has its own properties related to reachability and flow, producing different network behavior.

Overall, the DRS histograms have points that cluster tightly along a curve for smaller values of Δ, but eventually "pile" to the right as the maximum, or near maximum, number of nodes they may add is reached.

It has previously been discovered that Discrete Gaussian Exponential (DGX) distributions match well to many real world datasets [6], and fitting a DGX distribution to the curves obtained at small Δ values for the DRS, it can be seen from Figures 3, 4, and 5, that this distribution matches the DRS histogram at low Δ threshold values, as well.

An interesting insight can be discovered by plotting the negative of the log-log slopes from the power law distributions for each DRS histogram, such as those in Figure 2. The resulting curve for the Enron Email network can be seen in Figure 7. The negative of this log-log slope fits well to a log-normal curve, a property echoed in each of the other dynamic event networks analyzed, as well.

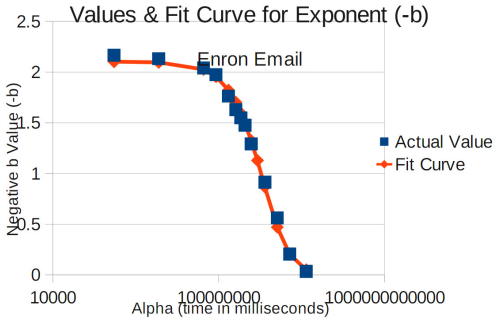

Fig. 7. Fitting a log-normal curve for the value of (-b), the negative exponent in the power law curve

3 The DRS-Gen Model

As a model, DRS-Gen focuses upon the generation and modeling of network dynamics and flow. A wealth of work in previous literature has focused upon the creation of static graph generators [4,7,10,17,18,20,28], and so, rather than repeat their work, DRS-Gen instead assumes that a base (static) graph structure is available. This structure could be the original network itself, without the dynamics and multi-edges, or it could be generated using any one of the many existing static network graph generators. For the experiments in this work, we use the original networks as a base, and generate dynamic behavior upon it.

The basis behind DRS-Gen is the shift in "weight" that occurs as the time interval Δ is increased. This shift can be seen for example in the DRS histogram plots of Figure 2, where a much larger number of points have collected to the right for the last graph in Figure 2, as compared to the preceding graph. This shift represents the fact that, as the time interval increases, more nodes are able to reach a larger section of the graph, giving them an increased DRS size. This property allows us to relate graph structure together with time. For any given time interval Δ, the associated DRS histogram essentially counts multiple small flows, separated by either time or graph structure. As the time interval increases, these smaller flows may join together to become a single larger flow, contributing toward the shift in mass, as two smaller flows are replaced by a new larger flow in the histogram.

To model this shift, DRS-Gen takes 5 parameters as input: a min and max time resolution Δ_{min} and Δ_{max}, a starting number of flows c, and two parameters, μ and σ which model the change in slope of a log-log power-law distribution fitted to the normalized DRS histogram, across time. It then proceeds in four basic steps.

1. First, we generate a series of c number of integers, representing the DRS sizes for a set of initial "base flows."
2. Next, we transform this series of integers into a series of small subgraph structures representing the flows.

3. We then search through the subgraphs, finding and combining subgraphs which overlap by choosing a time, δ_t, that represents the time differences between the occurrences of the flows.
4. Finally, we output the final generated graph. Initial flows are given random timestamps, and the δ_t values are used to calculate the timestamps for overlapping flows.

We describe these four steps in more detail in the following subsections.

3.1 Generating Flow Sizes

Given our minimum time resolution Δ_{min}, we want to generate a series of c integers. These integers represent the DRS size of our c initial "base flows". The series of DRS sizes should fit the appropriate normalized DRS histogram distribution, which we model using a power law curve, following the form:

$$pr[\mathbf{DRS}] = a(\mathbf{DRS})^{-b} \tag{1}$$

Where **DRS** stands for a particular DRS size. In this case, the exponent b represents the slope of the line arising within the log-log plot, as can be seen by taking the logarithm of both sides of Equation (1).

$$\ln(pr[\mathbf{DRS}]) = \ln(a) - b\ln(\mathbf{DRS}) \tag{2}$$

As the time interval (Δ) is increased, the amount of mass in the curve shifts to the right, resulting in a variation for b across time. Figures 6 and 8 showed that b varies across Δ and fits well to a log-normal distribution, which is represented by:

$$b = \frac{1}{\Delta\sigma\sqrt{2\pi}} e^{-\frac{(\ln\Delta-\mu)^2}{2\sigma^s}} \tag{3}$$

where μ and σ are the location and scale parameters of the fit log-normal curve.

Equation (1), together with Equation (3), can be used to relate the probability of seeing a DRS / flow size, against a particular time interval.

Since we wish to obtain a series of DRS sizes, **fSizes**, at base time resolution Δ_{min}, we substitute Δ_{min} for Δ into these equations, to obtain the probability distribution we wish to achieve. We find and sample from the inverse of the CDF of this probability distribution to obtain a similar distribution. The CDF becomes:

$$\begin{aligned} F_x &= \int_2^x ax^{-b} \\ &= \frac{a}{1-b}\left(x^{1-b} - 2^{1-b}\right) \end{aligned} \tag{4}$$

We invert it and obtain:

$$F_y = \left(\frac{(1-b)x}{a} + 2^{1-b}\right)^{\frac{1}{1-b}} \tag{5}$$

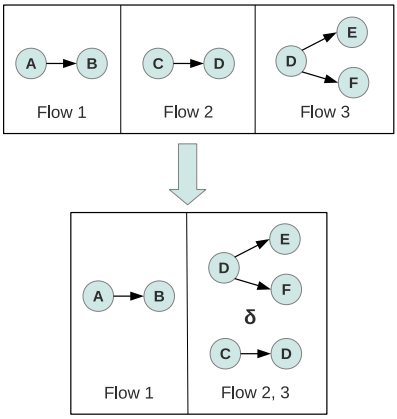

Fig. 8. Example of flows in F combining. Here, Flow 3 (of size 3 nodes) overlaps with Flow 2 (of size 2 nodes), and will therefore be combined, using a time difference δ between them, into a flow with 4 unique nodes.

Additionally, by assuming an approximate bound on the maximum DRS size, taken from the x intercept of the log-log plot shown in Equation (2) (at $\sqrt[b]{a}$), we can obtain a relationship for a using the pdf from Equation (1).

$$\begin{aligned} 1 &= \int_2^{\sqrt[b]{a}} ax^{-b} \\ &= \frac{\sqrt[b]{a} - 2^{1-b}a}{1-b} \end{aligned} \tag{6}$$

We can calculate the value for a from Equation (6) using Newton's method. After sampling from Equation (5), we then obtain our series of DRS flow sizes, **fSizes.**

3.2 Obtaining Subgraph Flow Structures

In order to obtain the subgraph flow structures, the series of integers found in the previous section must be transformed into a series of small subgraph structures, $F = [F_1, F_2, \cdots, F_c]$ of corresponding size. Drawn from the given base network, each subgraph will represent a single flow. There are multiple methods which can be used to create these small, static initial subgraph structures. We choose to use a simple variation (simply enforcing connected subgraphs) on the "Winners Don't Take All" method [29], which grows subgraphs by repeatedly choosing connected nodes either through random chance or through preferential attachment. The method used (random or preferential attachment) is randomly chosen at each step, as well.

3.3 Combining Overlapping Flows

With the series of base graph flows discovered, the next step is to combine overlapping flows. To do this, we assume a tentative ordering in time on F, and systematically search through the flows in F. For each flow, F_i, in F, we find all preceding flows, starting from F_1 and working our way up, which also overlap in graph structure. For every discovered overlapping pair of flows, we choose a time difference δ, using a probability function, that represents the amount of time that passes between their occurrence. These two flows are then combined (the time difference δ between them is noted) and the process continues. Figure 8 contains an example of this process, where Flows 2 and 3 overlap. A time difference δ is chosen between them, and they are combined. Pseudocode for this process is contained in Algorithm 2.

Given that we have found two overlapping simple flows of sizes S_1 and S_2, they will combine at some point in time, producing a single flow (of size S_3, where $max(S_1, S_2) \leq S_3 \leq S_1 + S_2 - 1$). The time difference between them is represented by δ. The two possibilities (whether the flows are combined or not) can be represented as two separate distributions: R, a distribution representing the combined state and having a probability of 1 for flow size S_3 and 0 for every other flow size, and U, a distribution representing the uncombined state and having a probability U_1 for flow size S_1, U_2 for S_2, and U_3 for S_3 (with U_1, U_2, and U_3 being discrete values of either 0, 0.5, or 1). As an example, in Figure 8, Flow 2 has a size of $S_1 = 2$ and Flow 3 has a size of $S_2 = 3$. When they combine, they produce a flow with size $S_3 = 4$. The probability distribution for the combined state, R, has probability 1 for size 4, and a probability of 0 for every other size. The distribution for the uncombined state, U, has probabilities 0.5 for size 2, 0.5 for size 3, 0 for size 4, as well as 0 for every other size.

The distance between these two possible distributions, and the modeled "true" distribution, T, of Equation (1) may be calculated. The likelihood of the flows combining may be found by comparing the distance between R and T with the distance between U and T. For distance comparison, we choose to use the Kullback-Leibler divergence (KL divergence).

$$\begin{aligned} D_{KL}(P||Q) &= \sum_{i \in S} P(i) \ln \frac{P(i)}{Q(i)} \\ &= \sum_{i \in S} P(i)(\ln P(i) - \ln Q(i)) \end{aligned} \tag{7}$$

From Equation (7), it can be seen that the KL divergence calculates a weighted distance between corresponding points on the log-log curve. This means that a distribution with a closer distance is more likely.

The probability values for the modeled distribution T can be found by using Equation (1), and are normalized.

$$T_i = \frac{P(S_i)}{\sum_{j \in S} P(S_j)} \tag{8}$$

Algorithm 2. DRS-Gen

```
1: procedure GENERATE(F)
2:     Initialize fSizes[ ]
3:     for i ← 1, c do
4:         r ← random number from Eq. 6
5:         fSizes[i] ← r
6:     Initilize F[ ]
7:     for i ← 1, c do
8:         Initialize K[ ]
9:         for j ← 1, fSizes[i] do
10:            K[j] ← new node using
11:              "Winners Don't Take All"
12:        F[i] ← K
13:    Initilize Deltas[ ][ ]
14:    for i ← 1, c do
15:        for j ← i − 1, 1 do
16:            if F[i] ∩ F[j] > 0 then
17:                if Deltas[i][j] isn't set then
18:                    δ ← random number from Eq. 13
19:                    Deltas[i][j] = δ
20:                    for k ← 1, c do
21:                        if Deltas[j][k] exists then
22:                            Deltas[i][k] = δ + Deltas[j][k]
23: end procedure
```

Substituting using Equation (1), we obtain

$$T_i = \frac{aS_i^{-b}}{\sum_{j \in S} aS_j^{-b}} = \frac{S_i^{-b}}{\sum_{j \in S} S_j^{-b}} \tag{9}$$

$$\left(where\ b = \frac{1}{\Delta\sigma\sqrt{2\pi}} e^{-\frac{(\ln \Delta - \mu)^2}{2\sigma^2}} \right)$$

When b is found using the parameters μ and σ, we obtain an equation relating the probability of certain sized flow occurring, to the time interval Δ.

To find the likelihood of combining, we calculate the relative closeness:

$$Pr[combining] = 1 - \frac{D_{KL}(R||T)}{D_{KL}(R||T) + D_{KL}(U||T)} \tag{10}$$

Substituting using Equation (7), we obtain Pr[combining] as:

$$1 - \frac{\sum_{i \in S} R_i(\ln R_i - \ln T_i)}{\sum_{i \in S} R_i(\ln R_i - \ln T_i) + \sum_{i \in S} U_i(\ln U_i - \ln T_i)} \tag{11}$$

Given that $R_1 = 1$, all other $R_i = 0$, and at least one of S_1, S_2, or $S_3 = 0$, we may simplify and combine with Equation (1), obtaining Pr[combining] as:

$$1 - \frac{b \ln S_1 + \ln(S_1^{-b} + S_2^{-b} + S_3^{-b})}{b \ln(S_1^{1+U_1} S_2^{U_2} S_3^{U_3}) + 2 \ln(S_1^{-b} + S_2^{-b} + S_3^{-b})} \tag{12}$$

where b is the log-normal curve shown in Equation (3), and all other values are constant. This probability varies across time, as Δ varies between Δ_{min} and Δ_{max}. A δ value, weighted by this distribution, can be picked using a series of approximations. The overall function and its integral can be approximated using Taylor series. Additionally, the area under the curve is calculated using the given value for Δ_{max}. Next, a random fraction of this AUC is chosen, and finally the appropriate δ for this AUC can be solved for numerically using Newton's method.

From this process, values for δ between overlapping flows are generated, with δ values fitting the dynamic distribution behavior discovered in Section 2.

3.4 Producing the Generated Graph

With the flow and timing information generated, these structures can be output to produce the final graph. Starting with the first flow F_1, a timestamp of 0 is assigned and output. Next, all other flows combined with F_1 are output, using their assigned time differences to produce their timestamps. If any flows remain, they are assigned a random timestamp, output, and their connected flows output as before. This process repeats until all flows have been output.

3.5 Parameter Fitting

From the steps in the overall algorithm, it is a simple extension to fit this generator to a known graph. First, a series of DRS histograms, at varying Δ, are calculated. Next, the c values for each histogram are normalized, producing a probability distribution. A power curve is fit to each distribution and the values for the power, b, extracted. From this series of b values, the appropriate values of μ and σ can be estimated by fitting a log-normal curve, and the generator may now be fit to the graph, using these parameter values.

Overall, this process allows a dynamic event graph to be generated, with dynamic reachability behavior fit to parameters learned from real world data.

3.6 Implementation and Analysis

Using this method, we implemented DRS-Gen and used the the captured DRS histograms for the Enron Email, Facebook wall post, and Twitter networks mentioned earlier and shown in Figures 2 and 3, to train (using the parameter fitting methods described in Section 3.5) a generative model capable of producing flows which imitate the properties of the original, real world graphs.

Figures 9, 10, and 11 show the resulting DRS histograms obtained through generation by the model, as compared to the original distribution. As can be seen,

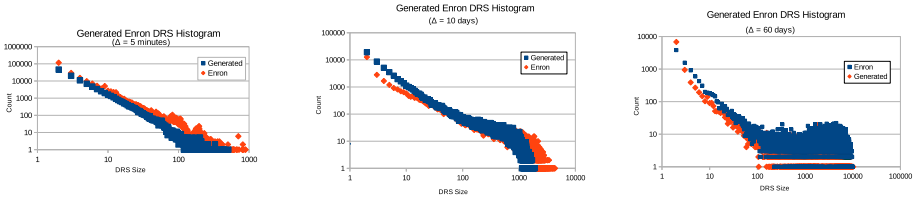

Fig. 9. Comparison of the DRS size histograms for both a graph generated using DRS-Gen, as well as the original Enron Email dataset, at increasing values for Δ. Both the tight fit of points, as well as the similarity in shape and dynamics emphasize the strength and quality of DRS-Gen's dynamic modeling results.

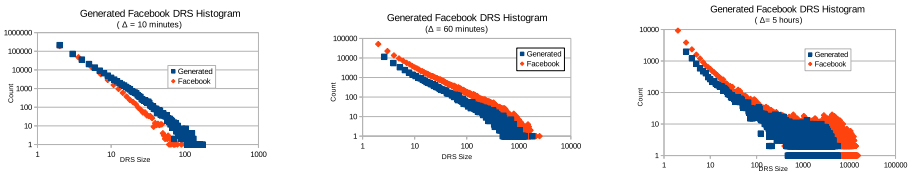

Fig. 10. Comparison of the DRS size histograms for both a graph generated using DRS-Gen, as well as the original Facebook dataset, at increasing values for Δ. Again, the strong similarity between both the original histogram and the generated, across time, helps to confirm the effectiveness of DRS-Gen's model.

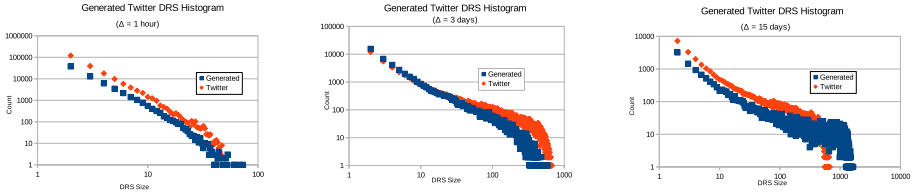

Fig. 11. Comparison of the DRS size histograms for both a graph generated using DRS-Gen, as well as the original Twitter dataset, at increasing values for Δ

both the distribution shapes, slopes, as well as the rates of change across time match extremely well to the original dynamic network distributions. Though each of the three original networks evolve at different rates, the graphs generated from our model manage to capture this evolution and fit the generated flows together in time such that the flow distribution and reachability match closely to the original curve.

A series of Quantile-Quantile plots are shown in Figure 12, comparing the original and generated distributions for the Enron Email, Facebook wall post, and Twitter graphs (at Δ =5 minutes, 10 minutes, and 1 hour respectively). The close fit to a straight y = x line further emphasizes the closeness of the generated vs. original distributions, and helps to confirm that our generative model is truly capturing the distribution and reachability of the original network.

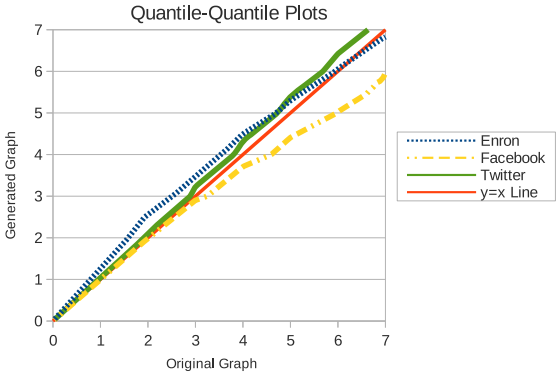

Fig. 12. Quantile-Quantile plots of the Enron Email, Facebook wall post, and Twitter graphs. The close fit to the y=x line emphasizes the closeness of the generated vs. original distributions.

These results help to confirm the effectiveness of our model, emphasizing its strength and ability to generating dynamic event network data, while capturing the dynamic reachability and flow properties of the original graph.

4 Related Work

Decades of research on graph theory has concentrated on studying fundamental properties of graphs and been successfully applied to the analysis and modeling of graph data and real world networks; however, researchers have mainly looked at static graph properties, leading to generative models that mimic static network structure [3,11,20]. In contrast, temporal graphs have been an active topic of research in only the last few years and the research has largely concentrated on dynamic "state" networks and ignored dynamic "event" networks [2,26,32].

A major focus of current research is on generative models that allow for the prediction of the slow evolution (long-term dynamics) of graph structure [12,16]. Previously introduced models for dynamic graphs include the Markovian Dynamic Graph models, which are random models where the graph structure at every time step t is dependent only on the structure at time $t-1$, and created according to random transition probabilities. In Edge-Markovian Dynamic Graphs [9], each edge at time step t is dependent only on its presence (or not) at $t-1$. There are fixed global birth and death rate functions, giving the probability of a new edge arising and an old edge dying. A variation on this model, where nodes are initially assigned a fixed position, and node distances affect birth and death rate values, was introduced in [13].

Despite their elegant formulation and ease of analysis, the Markovian Dynamic Graph models fail to capture many real-world network properties. For example, two general dynamic graph properties that have been observed are densification power laws, relating the number of nodes and edges of a graph over time to a power law distribution, as well as shrinking diameters across time [21]. To capture these discovered properties, new generative graph models were introduced.

On example is the Forest Fire model, where new links are formed by randomly choosing "ambassador" nodes and recursively following their links, linking to discovered nodes with a certain probability [21]. Other dynamic network properties recently discovered include the bursty-weight law, where edge weight additions were found to be bursty over time [27,19], and the relation between age of a node and its likelihood to attract new edges [20]. From these observations, new generative models such as the Butterfly graph model and Triangle-closing models were introduced [19,27]. Another recent graph generative model which accounts for numerous static as well as dynamic graph properties is RTG [1], based on the concept of random typing. In RTG, a set of keys have a probability distribution representing their likelihood to be typed. Every word randomly typed is a node label, and the stream of nodes typed are divided into source and destination pairs to create edges.

Dynamic processes on complex networks such as information diffusion and epidemiological processes have also been studied [5]. Epidemic models, such as the Susceptible-Infected-Susceptible (SIS) model [3], have been applied to the modeling of link cascades within blogs in [22,23]. Interestingly, even though the process is time-varying, the network in these models are usually considered static or changing very slowly. In contrast, recent work by Prakash et. al [30] analyzes virus propagation graphs by formulating them as an approximate nonlinear dynamical system. In [8], the authors utilize the spectral radius of the adjacency matrix for predicting the virulence of epidemics on static graphs.

Additionally, most current dynamic network generative models use the concept of abstract "timesteps" for their dynamics, which do not fit well with real world event graphs. Typically, for many models, each timestep label corresponds to a single event rather than a precise measure of time. This sequence of labels conveys the network dynamics. However, many real world networks instead have actual time values associated with each link, allowing for both evenly spaced flows as well as arbitrarily long pauses or bursts of activity. This timed behavior is an integral part of a dynamic network, and its oversight leaves a large area of important graph data and knowledge largely unexplored. The lack of time values upon edges also renders it difficult to calculate and compare DRS values from graphs generated by these algorithms to the original graphs, as Δ intervals cannot be easily mapped onto timesteps.

5 Summary

In this paper, we have introduced and analyzed a novel new property of dynamic event networks, their Dynamic Reachability Sets (DRS), across time. The DRS characterizes reachability within a graph across time, and connects to many important graph relationships such as network flow and latency, in addition to reachability. From this analysis, we have discovered several important new properties of dynamic networks, including a novel distribution pattern for the DRS histograms, related to a DGX distribution at small time intervals.

Additionally, we have made use of this newly discovered pattern by introducing a new generative graph model, DRS-Gen, based upon the DRS distribution

dynamics. DRS-Gen is capable of generating event network dynamics upon graphs, and particularly able to fit naturally to real world event networks, learning parameters that can capture and model dynamic flow and reachability across time. Dynamic graph models such as DRS-Gen can have many possible practical applications, including prediction of future graph evolution or behavior (such as in link or email thread prediction), and graph compression.

Implementing DRS-Gen and testing it on multiple networks, we find that the generated graphs closely matched the distributions and dynamics of the original networks they modeled, helping to emphasize DRS-Gen's use and effectiveness as a new dynamic event network graph generator, and a novel and potentially useful new tool for generative graph modeling and analysis.

Acknowledgements. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Work was also partially supported by the National Science Foundation under grant IIS-0917149.

References

1. Akoglu, L., Faloutsos, C.: RTG: A Recursive Realistic Graph Generator Using Random Typing. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part I. LNCS, vol. 5781, pp. 13–28. Springer, Heidelberg (2009)
2. Akoglu, L., Mcglohon, M., Faloutsos, C.: Rtm: Laws and a recursive generator for weighted time-evolving graphs. In: ICDM 2008 (2008)
3. Bailey, N.: The mathematical theory of infectious disease and its applications. Hafner Press (1975)
4. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
5. Barrat, A., Barthlemy, M., Vespignani, A.: Dynamical Processes on Complex Networks, New York, NY, USA (2008)
6. Bi, Z., Faloutsos, C., Korn, F.: The "dgx" distribution for mining massive, skewed data. In: KDD, pp. 17–26 (2001)
7. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Comput. Surv. 38 (June 2006)
8. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. ACM Trans. Inf. Syst. Secur. 10, 1:1–1:26 (2008)
9. Clementi, A.E., Macci, C., Monti, A., Pasquale, F., Silvestri, R.: Flooding time in edge-markovian dynamic graphs. In: PODC, pp. 213–222 (2008)
10. Erdös, P., Rényi, A.: On the evolution of random graphs. In: Publication of the Mathematical Institute of the Hungarian Academy of Sciences, pp. 17–61 (1960)
11. Fabrikant, A., Koutsoupias, E., Papadimitriou, C.: Heuristically Optimized Trade-Offs: A New Paradigm for Power Laws in the Internet. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, p. 110. Springer, Heidelberg (2002)

12. Goldenberg, A., Zheng, A.X., Fienberg, S.E., Airoldi, E.M.: A survey of statistical network models. Found. Trends Mach. Learn. 2, 129–233 (2010)
13. Grindrod, P., Higham, D.J.: Evolving graphs: dynamical models, inverse problems and propagation. Proc. of TRSA: Math, Phys. Engr. Sci. 466, 753–770 (2010)
14. Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.S.: The web as a graph: Measurements, models, and methods (1999)
15. Klimt, B., Yang, Y.: Introducing the enron corpus. In: CEAS (2004)
16. Kuhn, F., Oshman, R.: Dynamic networks: models and algorithms. SIGACT News 42, 82–96 (2011)
17. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E.: Stochastic models for the web graph. In: Proc. Found. of CS, pp. 57–66 (2000)
18. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tompkins, A., Upfal, E.: The web as a graph. In: PODS, pp. 1–10. ACM, New York (2000)
19. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic evolution of social networks. In: KDD (2008)
20. Leskovec, J., Chakrabarti, D., Kleinberg, J.M., Faloutsos, C.: Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 133–145. Springer, Heidelberg (2005)
21. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: KDD (2005)
22. Leskovec, J., Mcglohon, M., Faloutsos, C., Glance, N., Hurst, M.: Cascading behavior in large blog graphs: Patterns and a model. Technical report (2006)
23. Leskovec, J., McGlohon, M., Faloutsos, C., Hurst, M.: Cascading behavior in large blog graphs patterns and a model. In: SDM (2007)
24. Macropol, K., Can, T., Singh, A.: Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. BMC Bioinformatics 10, 283 (2009)
25. Macropol, K., Singh, A.: Scalable discovery of best clusters on large graphs. PVLDB 3(1), 693–702 (2010)
26. Macropol, K., Singh, A.K.: Content-based modeling and prediction of information dissemination. In: ASONAM (2011)
27. McGlohon, M., Akoglu, L., Faloutsos, C.: Weighted graphs and disconnected components: patterns and a generator. In: KDD, pp. 524–532 (2008)
28. Nickel, C.L.M.: Random Dot Product Graphs: A Model For Social Networks. PhD thesis, Johns Hopkins University, Maryland, USA (2006)
29. Pennock, D., Flake, G., Lawrence, S., Glover, E., Giles, C.L.: Winners don't take all: Characterizing the competition for links on the web. In: PNAS (2002)
30. Prakash, B.A., Tong, H., Valler, N., Faloutsos, M., Faloutsos, C.: Virus Propagation on Time-Varying Networks: Theory and Immunization Algorithms. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010, Part III. LNCS, vol. 6323, pp. 99–114. Springer, Heidelberg (2010)
31. Siganos, G., Faloutsos, M., Faloutsos, P., Faloutsos, C.: Power laws and the as-level internet topology. IEEE/ACM Trans. Netw. 11, 514–524 (2003)
32. Snijders, T.A., van de Bunt, G.G., Steglich, C.E.: Introduction to stochastic actor-based models for network dynamics. Social Networks 32, 44–60 (2010)
33. Tsourakakis, C.E.: Fast counting of triangles in large real networks without counting: Algorithms and laws. In: ICDM (2008)
34. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in facebook. In: WOSN 2009 (2009)