Technical Report CIS-TR-08-01, Department of Computer and Information Science, University of Oregon, September 2008

Evaluating Sampling Techniques for Large Dynamic Graphs

Amir H. Rasti, Mojtaba Torkjazi, Reza Rejaie
University of Oregon

Nick Duffield, Walter Willinger
AT&T Labs—Research

Daniel Stutzbach
Stutzbach Enterprises

1. EVALUATION: STATIC GRAPHS

In this section, we examine how the connectivity structure of the graph affects the efficiency of the RDS and MRW sampling techniques. Toward this end, we simulate both RDS and MRW techniques over several types of synthetic graph structures as well as one real graph. Note that we do not consider any of the synthetic graphs to be appropriate models of actual P2P systems or OSNs, but we use them in our experiments as examples for which we know the ground truth. At the same time, our choice of the models described below is not entirely arbitrary, but is motivated by the type of heterogeneities that we expect many of the real-world P2P networks and OSNs to exhibit.

Graph Models: We consider the following four synthetic graph models that range from very homogeneous to highly heterogeneous. This in turn allows us to explore the heterogeneity of a graph along two dimensions, namely node degree and clustering. We also use a real graph that is a full snapshot of the Gnutella ultrapeer overlay taken on 5/5/2008 at 3pm.

(i) Random graphs (ER): The well-known class of Erdös-Renyi random graphs [2], the simplest variety of random graphs where links between node pairs are inserted with probability p, independent of anything else.

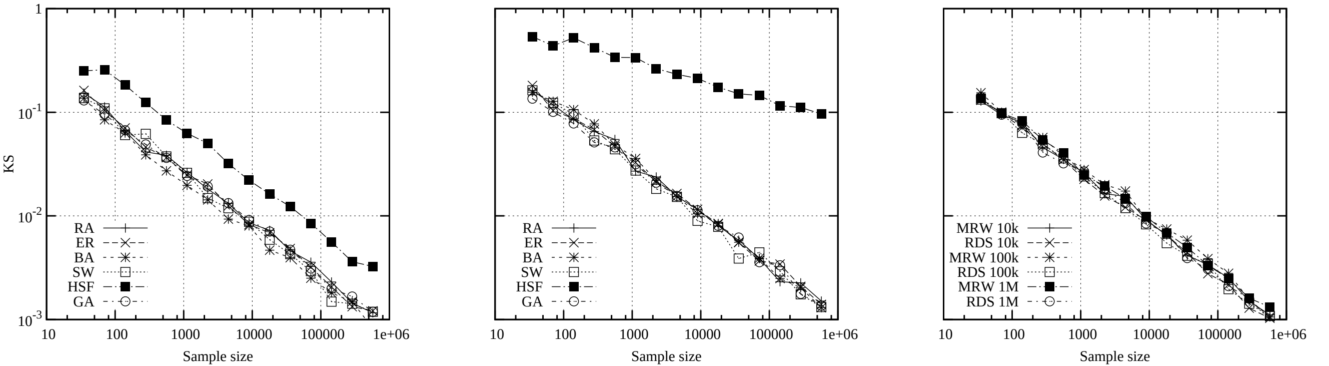
(ii) Small-world (SW): We consider here the "small-world" model proposed by Watts and Strogatz [11] who considered a one-parametric class of networks which interpolates between a regular ring lattice and a random graph without altering the number of nodes and links. This process generates graph structures with high clustering and small path lengths. $SW(p)$ has a single parameter p and increasing p reduces the degree of clustering in the graph.

(iii) Barabási and Albert (BA): Many real-world connectivity structures are heterogeneous in the sense that their nodes degrees or vertex connectivities exhibit high variability or, more specifically, follow a power-law distribution. To account for this type of heterogeneity, Barabási and Albert [1] proposed the class of scale-free models of the preferential attachment type, whereby graphs grow by addition of new nodes and links and where a newly arriving nodes connects with higher probability to an already highly connected node in the existing graph. Growing a graph in accordance to this *preferential attachment* mechanism can be shown to generate graph structures whose node degree distributions follow a power-law or are scale-free.

(iv) Hierarchical Scale-Free (HSF): In addition to the heterogeneity captured by highly variable node degrees, many real-world graphs also show heterogeneity in the sense of exhibiting clusters-within-clusters structure. A construction of a class of simple toy models of graphs that have power-law node degree distributions, significant clustering, and a pronounced hierarchical structure was given in [1]. In this article, Barabási *et al.* provide an iterative model for generating scale-free graphs with a hierarchical structure that we call *HSF* in this paper. An example is shown in Firgure 1 The graph $HSF(n, m)$ has n^{m+1} nodes. The HSF model takes two parameters: n and m : n denotes the size of the fully connected *cell* which is the building block of the graph, and m denotes the number of interations used to produce the graph. As shown in Figure 1, the construction proceeds by generating n cells of size n and connecting them in a certain way. Using the generated structure as a new cell, we repeat this process m times to obtain an HSF graph with m well-defined levels that manifest themselves as clearly visible clusters-within-clusters. This HSF graph can be shown to have a power-law node degree distribution and a high clustering coefficient, independent of the size of the network. As shown in Figure 4(e), both node degree and clustering have power-law distributions in HSF graphs. Also, it is shown that the node clustering is inversely proportional to node degree. In this class of graphs, node clustering is inversely proportional to node degree. This property provides a truely scale-free clustered graph as the average clustering coefficient does not change with the graph's order. [1]

(v) Gnutella (GA): Snapshots of the Gnutella ultrapeer topology, captured in our earlier work [9].

Figure 1(a) presents the KS error for degree distribution from samples collected by the RDS technique



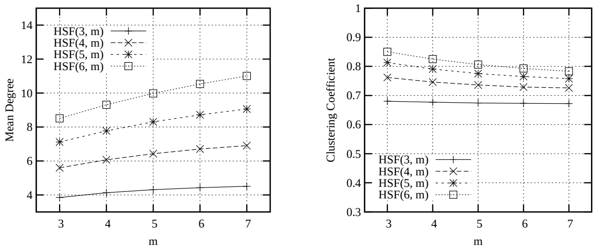
(a) Efficiency of RDS (b) Efficiency of MRW (c) Effect of Graph Size over Erdös–Rényi

Figure 1: Efficiency of RDS and MRW techniques in estimating degree distribution over different graph types and different graph sizes

over different graph types as a function of number of samples. Figure 1(b) shows the same results for MRW technique. To make the results more comparable, the number of vertices ($|V| = 390{,}625$) and edges ($|E| = 1{,}769{,}110$) in each graph are approximately the same.

These figures illustrate the following two important points. First, the accuracy of the RDS technique improves with the number of samples. The rate of improvement in accuracy across all graph types (*i.e.*, slope of the line) are similar. However, for a given number of samples, the accuracy of samples from HSF graphs are significantly lower. Second, the efficiency of the MRW technique is slightly worse, (e.g. the estimation error for RDS technique is on average around $2.2 \cdot 10^-3$ less than the MRW technique across different graphs except HSF) but follows a similar trend over all graphs except HSF. MRW sampling not only exhibits a significantly lower efficiency over HSF compared to other graphs, but the rate of improvement in accuracy with the sample size is much worse for HSF. Figure 1(c) presents the efficiency of both RDS and MRW sampling over Erdös–Rényi graphs with different sizes. This figure demonstrates that the accuracy of both sampling techniques over the Erdös–Rényi graphs depends on the absolute number of samples regardless of the graph size. Sampling over BA and SW graphs as well as Gnutella snapshots exhibit the same behavior.

To further investigate the interactions between the sampling techniques with HSF graphs, we examine the efficiency of both RDS and MRW techniques over HSF graphs with different parameters (n, m). Changing these parameters affects the mean degree and clustering coefficient of the HSF graph but its degree distribution remains power-law. Figure 6 summarizes the effect of n and m on properties of the HSF graph. In summary, for a given cell size (n), increasing the number of levels of the hierarchy (m) increases the mean degree. Clustering coefficient also initially decreases but as it converges to a value around 0.7-0.8 as the graph order (m) increases. The rate of changes with m is higher for larger values of n.



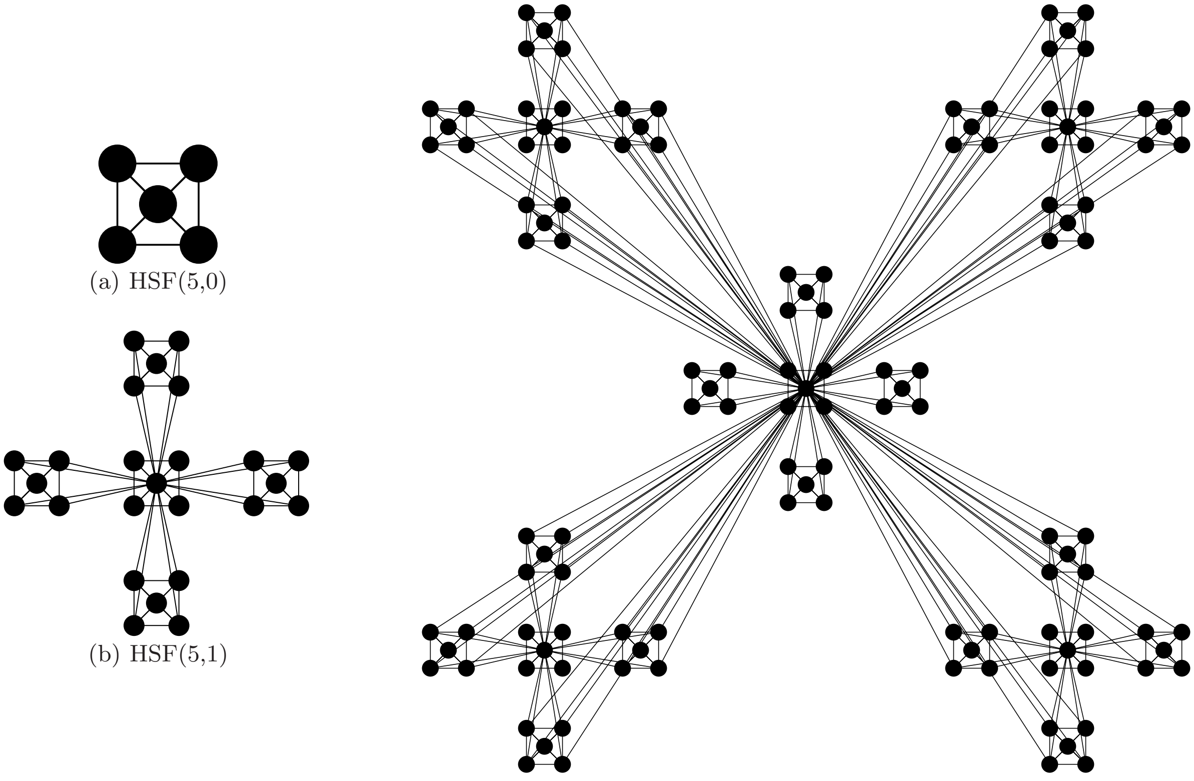
(a) HSF Mean Degree (b) HSF Clust. Coef.

Figure 6: Specifications of HSF graphs for different parameters

Figure 3(a) depicts the efficiency of RDS and MRW techniques over HSF(n,m) graphs with different parameters as a function of sample size. This figure illustrates that the efficiency of the RDS technique is minimally affected with the change of HSF parameters but the impact on MRW sampling is quite pronounced. We further change the structure of the HSF graph by performing random degree-preserving rewiring of a certain portion of the edges in the graphs.

Table 1: Effect of Edge Shuffling on the clustering of HSF(5,7)

% Shuffled	Mean Deg.	Clust. Coeff.
0	9.06	0.758
1	9.06	0.727
5	9.06	0.615
10	9.06	0.502
50	9.06	0.127



(a) HSF(5,0)

(b) HSF(5,1)

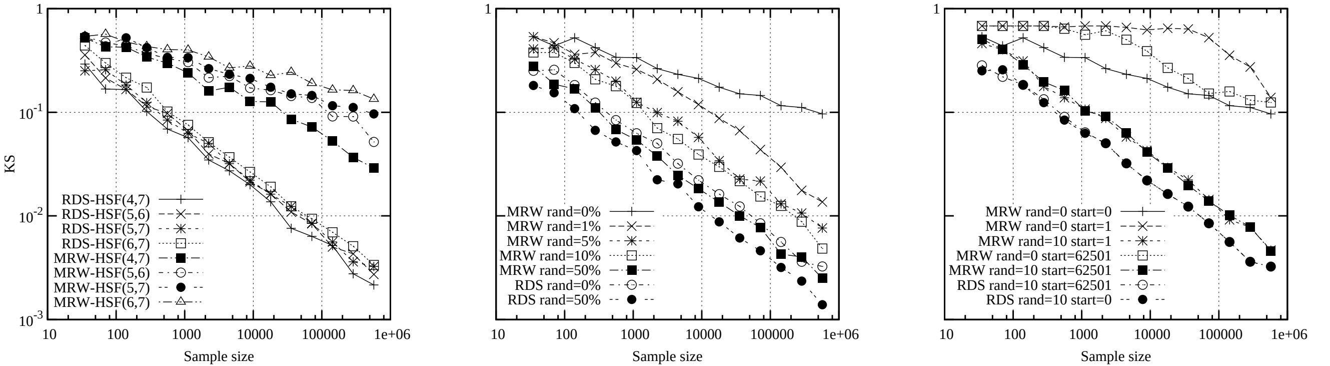
(c) HSF(5,2)

Figure 2: Building HSF graph. Figures borrowed from [1]

Table 1 summarizes the effect of shuffling 0%, 1%, 5%, 10% and 50% of edges in an HSF(5,7) graph. Figure 3(b) shows the efficiency of MRW sampling technique over HSF(5,7) when 0%, 1%, 5%, 10% and 50% of edges are shuffled (randomly rewired). We have also presented the results for the RDS technique over the same graph when 0% and 50% of edges are shuffled for comparison. The figure demonstrates that even a small percentage of shuffled edges dramatically improves the efficiency of the MRW technique. Figure 3(c) presents the sensitivity of the MRW sampling technique to the choice of starting node for random walk over HSF(5,7) graphs with 0% and 10% shuffled edges [1]. The figure indicates that the behavior of MRW sampling over HSF graph is very sensitive to the choice of starting point of the walk. More specifically, the lower the degree of the starting node, the more likely the walker is trapped within a cluster for a period of time. In fact, for some starting points, the accuracy of sampling does not improve with sample size until it passes a large threshold. However, adding some randomness to the HSF graph (by shuffling the edges) eliminates the sensitivity of MRW's behavior to the starting point as shown in Figure 3(c).

The reported difference in the efficiency of RDS and MRW sampling techniques over the HSF graph is caused by the combined effect of skewed distribution of node degree and higher clustering in the graph More specifically, at each step of the random walk in the RDS sampling, all neighbors of the current node can be selected with equal probability. A walker may walk across nodes within a highly clustered cell for a while but there is also a reasonable chance that it leaves a cell to explore other cells. A trapped walker within a low degree cluster keeps collecting redundant samples of low degree nodes which in turn degrades the efficiency of the sampling. In the MRW technique, the probability of selecting a neighbor as the next hop is inversely proportional with the ratio of its degree to the degree of current node when this ratio is greater than one. Given the power-law distribution of node degree, when a walker is within a cell where all nodes are clustered and have the same small degree, the only way to leave the cluster is to select a high degree node in the higher level of the hierarchy. However, the degree of the node in the higher level is an order of magnitude larger, and thus the probability

[1] We examined the behavior of MRW from many starting points at different levels of the hierarchy in HSF. The presented results in Figure 3(c) represent the most common behavior across different starting points.



(a) HSF graphs (b) HSF graphs with edge shuffling (c) HSF graphs with edge shuffling and different starting points

Figure 3: Efficiency of RDS and MRW in estimating degree distribution over HSF graphs

of selecting that node (*i.e.*, leaving the current cluster) is very small. As a result, a walker may become trapped within a single cluster for a long time.

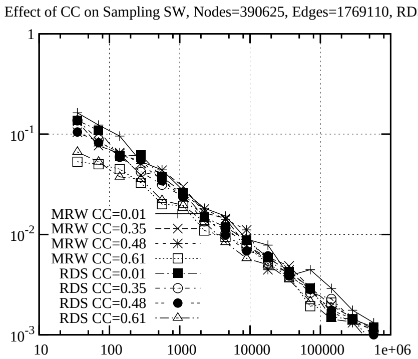


Figure 7: Efficiency of RDS and MRW over $SW(p)$ with different levels of clustering

Rewiring edges of the graph changes the strict structure of the graph and provides more opportunities for a walker to leave individual clusters. Note that the BA graph does not have a similar challenge for the sampling techniques despite its power-law distribution of node degree because its random connectivity results in homogeneous clustering across the network (i.e., clustering coefficient of a node with degree k is independent of k) and always provides opportunity for a MRW walker to leave any given neighborhood. We also examined the performance of both sampling techniques over $SW(p)$ graph with different levels of clustering, and observed that the level of clustering for $SW(p)$ does not have any significant impact on the efficiency of RDS and MRW sampling techniques as shown in Figure 7. This figure offers a strong evidence that high degree of clustering alone does not degrade the efficiency of either sampling techniques. We should note that decreasing p, the single parameter of the SW model, in addition to increased clustering, also causes more uniform degree distribution. This can explain the fact that in Figure 7, highly clustered graphs show better sampling results.

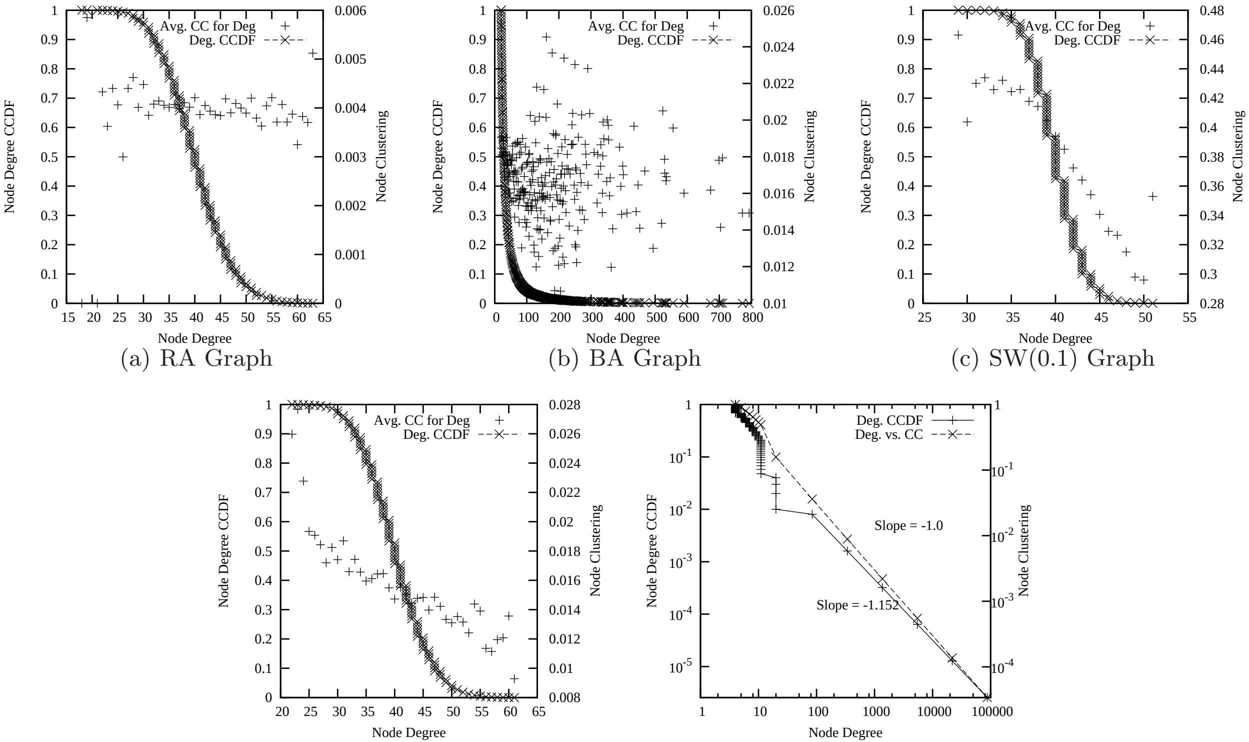
To reliably determine the underlying causes for the degraded efficiency of MRW sampling over HSF graphs, we have derived "cover time" for both random (RDS) and MRW walkers over a range of graph types with different parameters. Table 2 summarizes the results and includes the clustering coefficient (CC) for individual graphs [2]. For RA, BA and SW models, there average degree is 20 and each experiment is repeated 40 times with different random seeds. Table 2 confirms our explanation and shows that MRW technique usually exhibits a longer cover time. However, the MRW technique is significantly more sensitive to the combination of high clustering and skewed distribution of degree. *In summary, our results in this section concluded that both RDS and MRW are equally capable of efficiently estimating key node properties over static graphs with various degree distributions and various levels of clustering. However, the efficiency of MRW sampling is significantly degraded over graphs that exhibit both highly skewed degree distribution and a high level of clustering due to its inability to effectively explore different regions of the graph.*

1.1 Evaluation Over OSN-like Snapshots

To examine both techniques over more realistic snapshots of OSNs, we have used the snapshots of four OSNs that were captured by Mislove et al. [5] and are publicly available online [3]. As reported by Mislove *et al.*, between 0% (for Orkut) to 40% (for Flickr) of links in these snapshots do not have the reverse edge. We

[2] We note that cover time is the number of steps that are required for a walker to visit all nodes of a given graph.

[3] http://socialnetworks.mpi-sws.mpg.de/data-imc2007.html



(a) RA Graph (b) BA Graph (c) SW(0.1) Graph

(d) SW(0.5) Graph (e) HSF(5,6) Graph

Figure 4: Degree distribution and average node clustering for different graph types

Table 2: Cover Time for MRW and RDS.

Graph	# nodes	CC	MRW	RDS
RA	10K	$2.0 \cdot 10^{-3}$	144K	147K
BA	10K	$1.1 \cdot 10^{-2}$	261K	175K
SW(0.8)	10K	$2.0 \cdot 10^{-3}$	144K	141K
SW(0.5)	10K	$1.3 \cdot 10^{-2}$	132K	129K
SW(0.1)	10K	0.38	121K	119K
HSF(5,5)	3.1K	0.78	20M	414K
HSF(5,6)	15.6K	0.77	1,189M	3.5M

have added these reverse edge in order to generate a bi-directional graph for evaluating our sampling techniques [4]. We note that these revised graphs may not represent any real world OSNs since adding a small percentage of links could significantly change properties of a graph. However, we use these graph as a rough estimate for OSN connectivity structure and believe that they are more realistic than HSF graph.

Figure 8 depicts the efficiency of RDS and MRW technique over the revised snapshots of four OSNs. We also included the clustering coefficient of the revised graphs. These results show that RDS outperforms MRW in all graphs. The efficiency of RDS and MRW over YouTube, Orkut and LiveJournal snapshot is very similar to their behavior over HSF graph with edge shuffling (in Figure 3(b). Our findings can be explained by the fact that these snapshots exhibit skewed distribution of node degree coupled with high clustering as reported by Mislove *et al.*. [5]. The relatively lower efficiency of RDS over the Flickr snapshots is likely to be due to the large diameter of the Flickr snapshot coupled with the fact that only a small fraction (12.24%) of the whole network has been captured in the snapshot. In summary, these results suggest that RDS is a promising technique to sample OSN-like graphs.

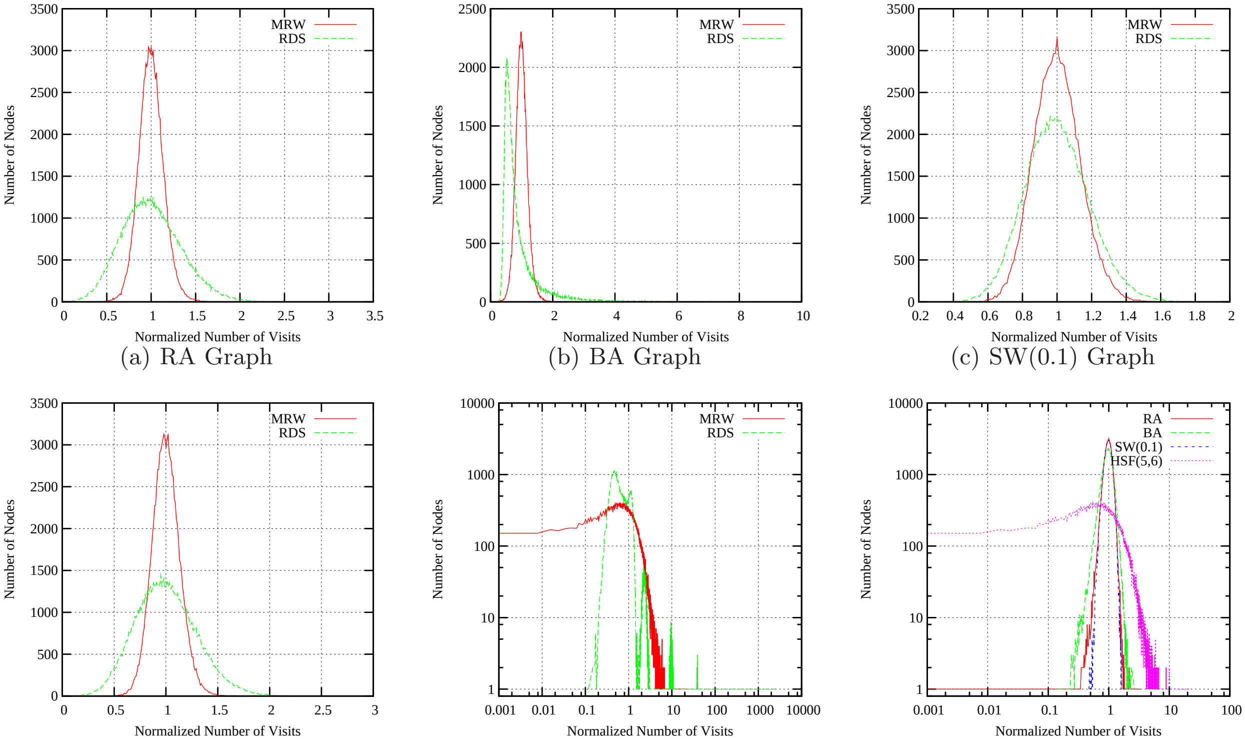
2. EVALUATION: DYNAMIC GRAPHS

This section explores high-fidelity measurements using RDS and MRW on dynamic graphs via simulation. We will show that a key trade off for accuracy is the number of parallel samplers to employ. Furthermore, we will show that there is an upper-bound on accuracy regardless of the level of parallelism and number of samples gathered. Finally, we will demonstrate that peer churn characteristics are a driving force in determining the upper-bound on accuracy.

2.1 Simulation Environment

In this section, we examine the behavior of the RDS

[4] Unfortunately removing the directed edges results in the fragmentation of the graphs and was not a viable option.



(a) RA Graph (b) BA Graph (c) SW(0.1) Graph

(d) SW(0.5) Graph (e) HSF(5,6) Graph (f) MRW on different graphs

Figure 5: Distribution of node visits for different graph types

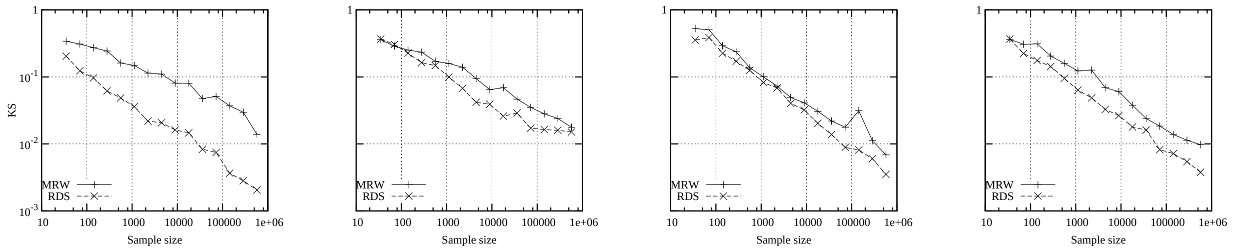
and MRW techniques over dynamic graphs using simulation. Characterizing the dynamics of OSNs is still a challenging area of research. In this section, we focus on simulating P2P overlays since their dynamics are well understood [6].

In our prior work [8], we developed a session-level simulator, called *psim*, that models peer arrivals, departures, latencies, and neighbor connections. The latencies between peers are randomly selected from values in the King data set [4]. Peers use a peer discovery mechanism to learn about other participating peers in the session to establish new connections. We have examined several central peer discovery mechanisms and since their impact on our analysis is negligible, we use a bootstrap node with FIFO strategy in our simulation for peer discovery [3]. Peers contact a bootstrap node which returns a list of the last n peers that contacted the bootstrap node, where n is the maximum peer degree. Peers have a target minimum number of connections that they attempt to maintain at all times. Unless otherwise noted, nodes attempt to maintain at least 30 connections. Whenever a peer loses a neighbor and the number of its connections drops below the target, it uses the discovery mechanism to learn about other peers and establish additional connections. Each peer limits its number of neighbors to 60. We assume connections require a 3-way handshake before the connection is fully established (*e.g.*, TCP), and that a connection request to a departed peer will time out after 10 seconds. To query a peer for a list of neighbors, the sampling node must establish a TCP connection, submit its query, and receive a response. The query times out if no response is received after 10 seconds.[5] Starting from an empty overlay, we run the simulator for a warm-up period to reach steady-state conditions before gathering samples.

The *psim* tool simulates churn by controlling the distributions of peer inter-arrival intervals and peer session lengths. A new peer selects its session length from a predefined distribution and departs when its session expires. New peers arrive according to a Poisson process, where the mean peer arrival rate combined with the session length distribution yield a desired mean population size in steady state. In this paper, the simulations use a population size of 100,000 peers.

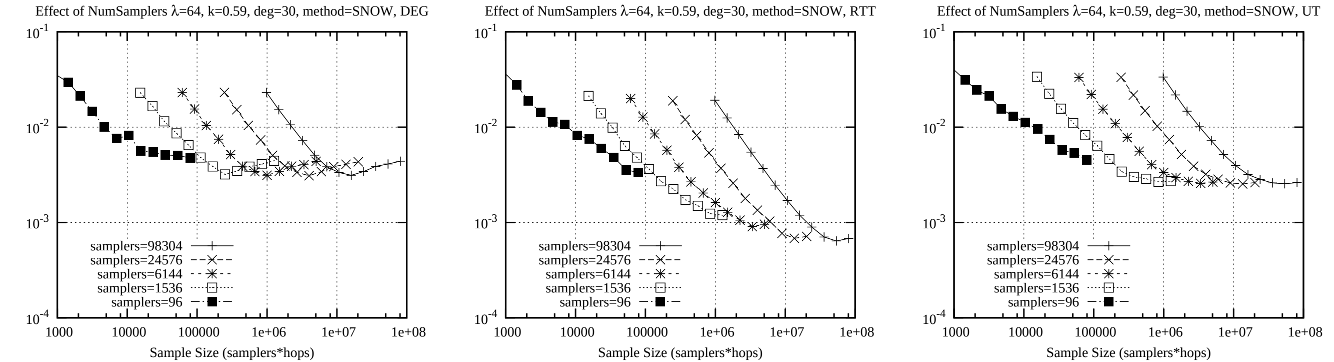
In this paper, we use Weibull distribution to model the peer session lengths based on our earlier empirical observations [6]. Weibull distributions (shape k, scale λ) provides a suitable model of peer session lengths, representing a compromise between the exponential and Pareto distributions.

[5] The value of 10 seconds was selected based on our experiments in developing a crawler for the Gnutella network.



(a) YouTube, CC = 0.074 (b) Flickr, CC = 0.169 (c) Orkut, CC = 0.168 (d) LiveJournal, CC = 0.269

Figure 8: Sampling online social networks: *YouTube, Flickr, LiveJournal, Orkut*



(a) Effect of Number of Samplers - Sampling Node Degree (b) Effect of Number of Samplers - Sampling RTT from Hypothetical Sampling Host (c) Effect of Number of Samplers - Sampling Node Uptime

Figure 9: Effect of Number of Samplers on Sampling of Each Parameter

Intuitively, if there is any bias in the samples, the bias will be tied to some property that interacts with the walk. In a nutshell, if there were no properties that interacted with the walk, then the walking process behaves as it would on a static graph, for which we have a proof of accuracy from graph theory. Therefore, we are only worried about properties which cause the walk to behave differently. We identify the following three fundamental properties that interact with the walk:

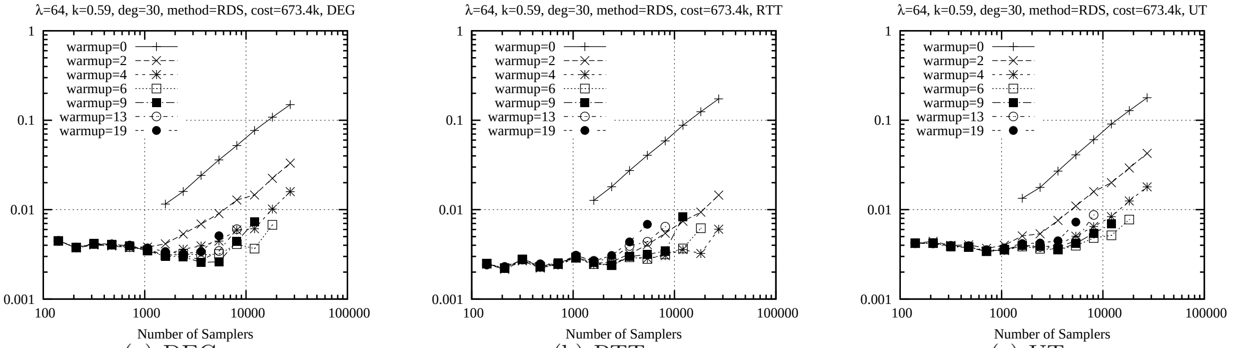
- *Node Degree*: The degree of individual node in the graph determines the probability that a node is visited by a random walker.
- *Session lengths*: The dynamics of peer participation derives the evolution of the graph with time. If the walker is more likely to select either short-lived or long-lived peers, there will be a bias correlated with session length.
- *Query latency*: In a static environment the only notion of time is the number of steps taken by the walk. In a dynamic environment, each step requires querying a peer, and some peers will respond more quickly than others. This could lead to a bias correlated with the query latency. In our simulations, we model the query latency as twice the round-trip time between the sampling node and the peer being queried.

2.2 Sequential versus Parallel Sampling

Sampling via random walks lends itself well to parallelism. To collect n samples, the sampling tool has a spectrum of options:

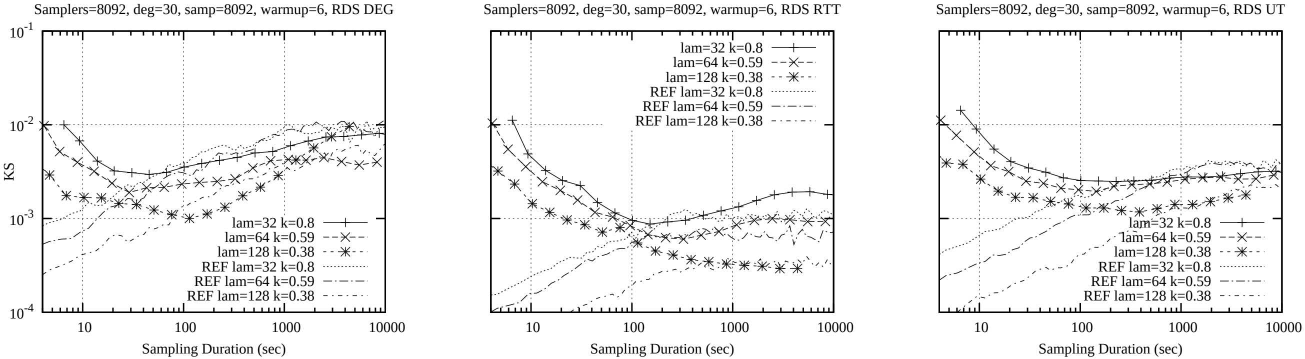
1. *Sequential*: It can make a short warm-up walk, then sample the next n hops sequentially.
2. *Parallel*: It can initiate n walks in parallel, each of which goes through a warm-up, then collects 1 sample.
3. *Hybrid*: It can take a hybrid approach, where it initiates $k \in [1, n]$ walks in parallel, each of which goes through a warm-up, then samples the next $\frac{n}{k}$ hops sequentially.

Throughout this paper, we refer to each one of the parallel walkers as a *sampler*. The sequential approach has the lowest bandwidth and processing power cost to gather n samples, as it only performs one warm-up. However, the parallel approach will gather the samples



(a) DEG (b) RTT (c) UT

Figure 10: Sensitivity to warmup length in RDS sampling. Graph size: 100K nodes, target degree: 30, churn parameters: $\lambda = 64$, $k = 0.59$



(a) DEG (b) RTT (c) UT

Figure 11: Effect of Churn parameres on RDS performance, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092

the fastest. Because the graph is changing as the samples are collected, speed can translate into greater accuracy. Consequently, giving up some samples in exchange for faster execution can sometimes result in an overall improvement in accuracy.

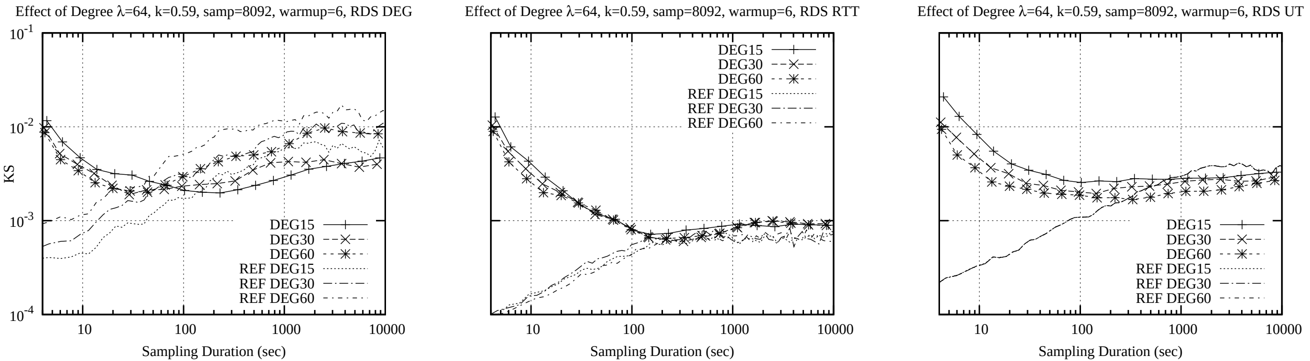
Figures 13 and 14 demonstrate this effect. The x-axis demonstrates the total cost of gathering the samples, in the number of peers visited by the walk (directly proportional to $bandwidth \cdot time$). Each line in the figures represents a different level of parallelism. Each line has a general downward trend, demonstrating that increasing the sample size also increases the accuracy (decreases error). However, each line has a plateau, where increasing the sample size *decreases* accuracy (increases error). This plateau is the point where the penalty for extra time required to gather the additional samples outweighs the benefits of having additional samples. Because parallelism reduced the time cost, increased parallelism improves the plateau. In other words, *parallelism is expensive but is needed to break past an accuracy plateau.*

2.3 Limit of Accuracy

Even with pure parallelism, collecting more samples requires more time due to the increasing probability of encountering a string of slow or departed peers (that leads to timeout). To explore the limits of accuracy, we use the notion of a *performance front*: the best (mean) accuracy achievable across the set of possible levels of parallelism, given a fixed time budget in which to collect all the samples.

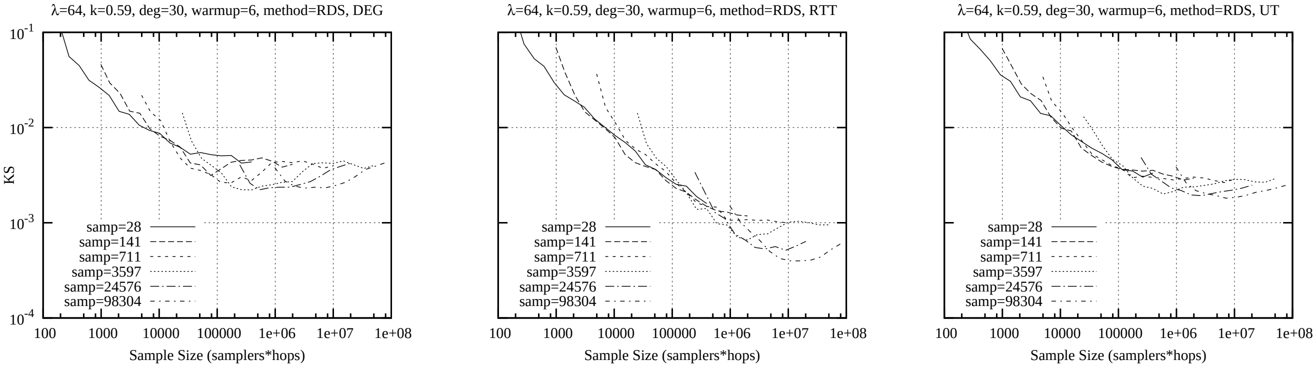
Figure 15 presents the performance fronts for RDS technique. Again, we see a pattern where the sampling accuracy improves from left-to-right. In Figures 15(a), the accuracy degrades on the far right of the graph.

In addition to the sampling results, each graph includes a reference line, labeled “SnapKS”. The reference line shows the KS error between a perfect snapshot gathered at time 0 (when the sampling process was initiated) and another snapshot gathered at time x. It demonstrates how much the distribution is changing while the sampling process occurs. At time 0, the KS error is also 0, since the distributions have not changed at all. As time passes, changes in the system alter the distributions slightly causing the KS error to increase as the system slightly drifts away from the state at time



(a) DEG (b) RTT (c) UT

Figure 12: Effect of Degree parameres on RDS performance, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092



(a) Degree (b) RTT (c) Uptime

Figure 13: Effect of warmup phase (w=6) on the efficiency of RDS sampling. Graph size: 100K nodes, target degree: 30, churn parameters: $\lambda = 64$, $k = 0.59$

0. Because the systems are in steady state, the reference line will eventually level out, which is visible in Figure 15(b).

The intersection of the sampling lines and the reference line represent a bound on the accuracy. Once the sampling line crosses the reference line (labeled as "SnapKS"), adding more samples will not improve the accuracy because the system has changed too much. This effect is visible in Figure 15(a), where the best accuracy is observed at around $x = 40s$. After that point, the accuracy only degrades.

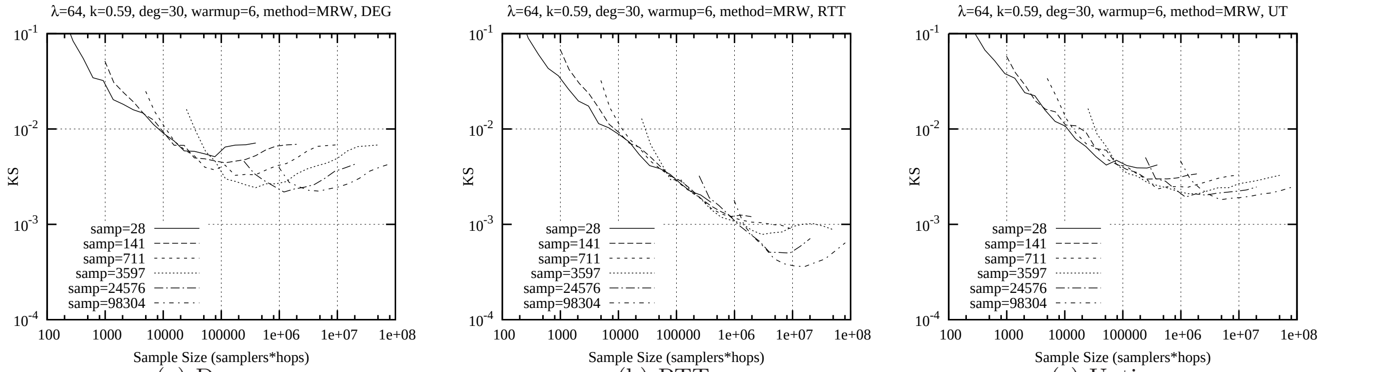
In summary, *system dynamics ultimately limits accuracy, regardless of the number of samples or level of parallelism.* It is worth noting that the KS value across all the scenarios in these figures are less than 0.01. This means that the error in each scenario is less than 1 percent which is adequate for most purposes.

2.4 Effect of Churn and Degree

In previous subsection, we showed the evolution of the system during the collection of samples is a key factor that limits accuracy. There are two important parameters that directly affect the evolution of node properties over time as follows: *(i)* peer churn, *(ii)* node degree. This subsection explores the effect of churn and target node degree on the maximum level of accuracy.

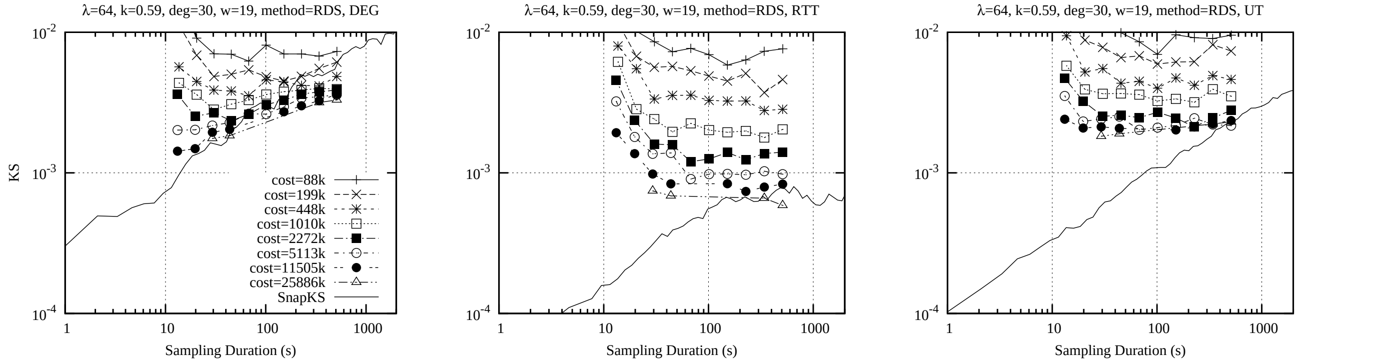
Effect of Churn: To explore the effect of churn, we consider the Weibull distribution for session length with a variety of parameters(λ, k) as follows: *(i)* λ=32, k=0.8, *(ii)* λ=64, k=0.59, and *(iii)* $\lambda = 128$, $k = 0.38$. In summary, decreasing λ reduces the median session length and decreasing k (for the fixed λ) make the distribution more skewed. Both the median and average session length monotonically increase (*i.e.*, churn rate decreases) from scenario *(i)* to scenario *(iii)*.

Each dotted line in Figures 18 and 19 present the quality of collected samples using RDS and MRW with a fixed number (8092) of samplers and one of the above three settings for churn. We have shown the results for estimating degree, RTT, and uptime. Note that the sampling budget (or cost) is increased with time as each sampler collects more samples over time, *i.e.*, the sam-



(a) Degree (b) RTT (c) Uptime

Figure 14: Effect of warmup phase (w=6) on the efficiency of MRW sampling. Graph size: 100K nodes, target degree: 30, churn parameters: $\lambda = 64$, $k = 0.59$



(a) Degree (b) RTT (c) Uptime

Figure 15: Efficiency of RDS, target degree = 30, churn parameters: $\lambda = 64$, $k = 0.59$, Warmup = 19 hops)

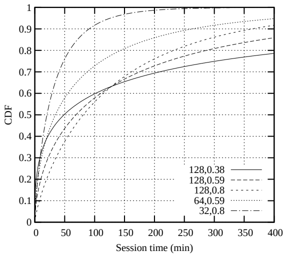


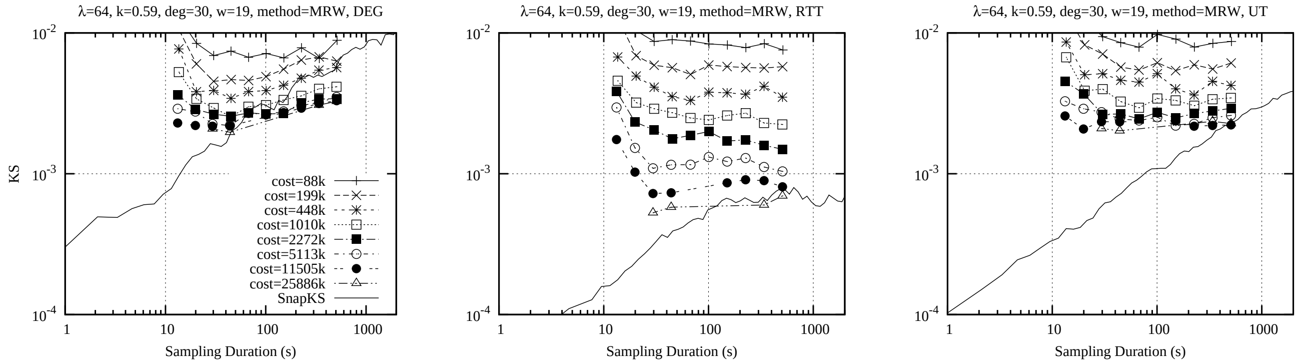
Figure 17: Weibull distribution (λ, k)

pling budget is not fixed. In each figure, each diagonal line (labeled REF) also presents the reference error for a particular setting for churn, *i.e.*, the difference (KS) in the distribution of the corresponding node property between the start of sampling (time 0) and time t. These figures show that as the churn increases, the reference error line shifts higher, *i.e.*, over a given period of time, the underlying distribution undergoes larger changes. They also show that the quality of collected samples with the RDS method over time t improves as churn rate decreases. In other words, *shorter peer lifetimes decrease the maximum sampling accuracy.*

Effect of Target Node Degree: Figure 18 presents the effect of target node degree for different values (15, 30 and 60) on the quality of collected samples by 8092 RDS samplers when churn parameters are $\lambda = 64$ and $k = 0.59$. We have shown the quality of samples for estimating degree, RTT and uptime, and included the corresponding reference error lines in each figure. These results show that *the target node degree has no effect on the accuracy of RTT or uptime estimates*. It does have a notable effect on the degree estimates, primarily because higher target degree increases the fluctuations within the steady state as shown by the reference lines in Figure 20(a). In other words, departure or arrival of a single node will affect a llarger population when target degree is larger.

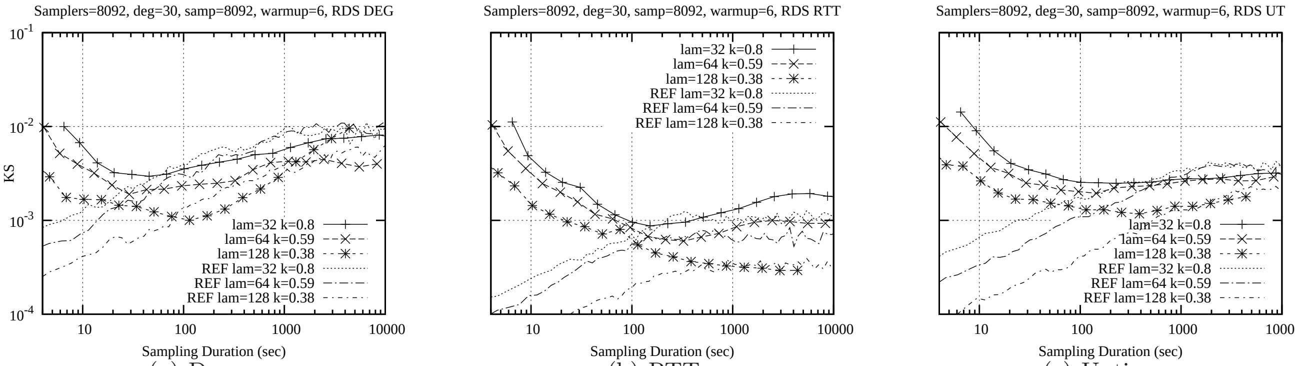
3. EMPIRICAL EVALUATION

We have also implemented both the RDS and MRW



(a) Degree (b) RTT (c) Uptime

Figure 16: Efficiency of MRW, target degree = 30, churn parameters: $\lambda = 64$, $k = 0.59$, Warmup = 19 hops)



(a) Degree (b) RTT (c) Uptime

Figure 18: Effect of Churn parameters on RDS efficiency, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092, Warmup=6

methods to sample widely deployed P2P and OSN systems in order to empirically validate the methods. Toward this end, we focus on Gnutella and Friendster[6] as our target P2P and OSN systems, respectively. We select these systems because they both have a well-defined API as well as large and active populations of users. We recall that our sampling methods assume an undirected connections among nodes. While this condition is easily satisfied for most P2P overlays, the connectivity among users in some OSNs is directional, *i.e.*, user A lists user B as a friend but the reverse friendship may not exist. To apply our sampling technique to OSNs, we focus on OSNs where pairwise relationship between users is bidirectional[7]. Several OSNs such as Orkut[8] and Friendster have an undirected connectivity among users. As part of our future work, we are designing techniques that can derive unbiased estimates from directed graphs.

The main challenge for the empirical validation of our sampling method is the lack of a reference snapshot of a desired node property with which to quantify the error. We take advantage of system-specific opportunities to extract a relatively accurate reference to use as reference.

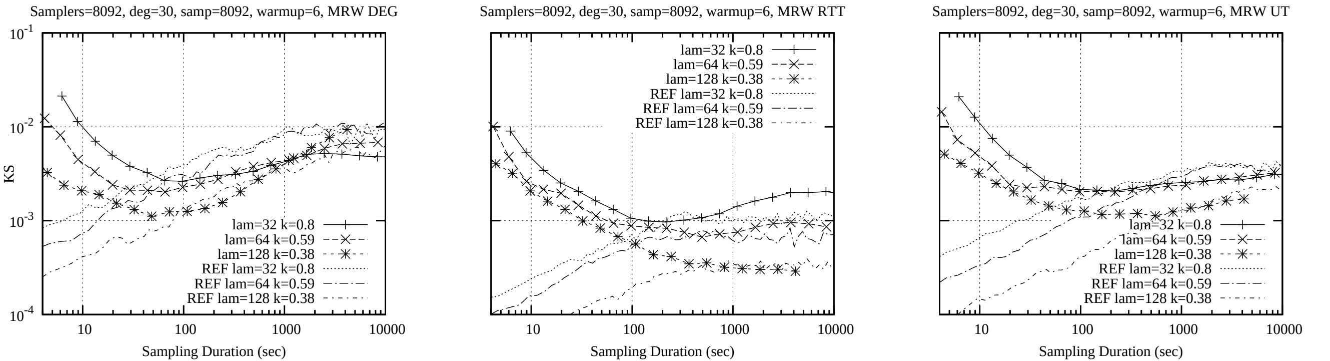
3.1 Validation over Gnutella

To evaluate the efficiency of our sampling techniques over Gnutella, we incorporate RDS technique into our sampling tool, ion-sampler [7]. We concurrently started 1000 RDS and 1000 MRW samplers while using Cruiser [10] to collect complete back to back snapshots of the top-level overlay every 7 minutes. Each sampler takes a 2000-step walk among ultrapeers in Gnutella network to collect samples of node degree. The combination of 1000

[6] http://www.friendster.com/

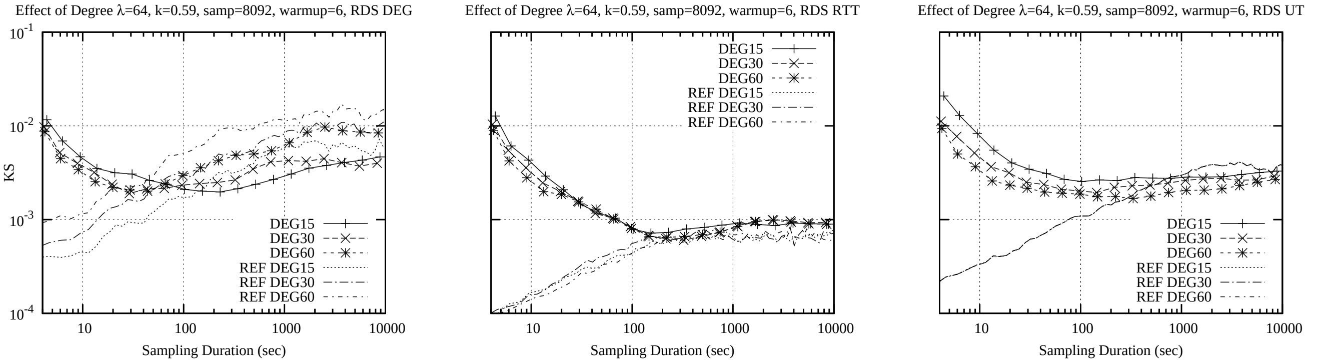
[7] It is worth noting that our sampling method can also handle OSNs (such as LiveJournal) whose API provides the following information for a given user u: the list of the u's friends (u's outgoing edges), and the list of other users in the system that present u as a friend (u's incoming edges). We can use this information to trivially transform a graph into an undirected graph.

[8] Several OSNs, including Orkut, have agreed to support OpenSocial API, but at this point neither the API is clearly defined nor many of those OSNs actually support the API. This eliminated Orkut as a good choice for this study to measure.



(a) Degree (b) RTT (c) Uptime

Figure 19: Effect of Churn parameters on MRW efficiency, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092, Warmup=6



(a) DEG (b) RTT (c) UT

Figure 20: Effect of Target node degree on RDS efficiency, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092, Warmup=6

samplers with 2000 walk length enable us to emulate various combination of smaller number of samplers and shorter walk length.

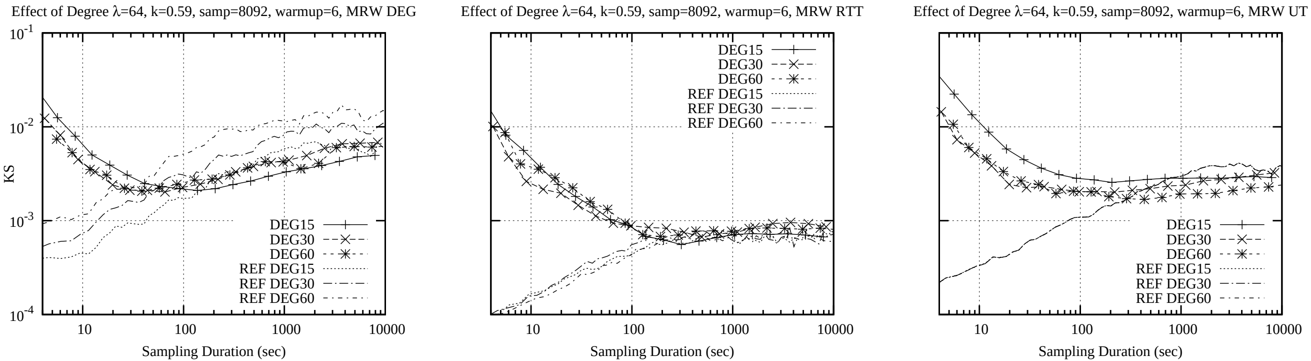
Figures 22(a) depicts the efficiency of both RDS and MRW techniques in estimating distribution of node degree with warmup phase of 5 hops. Figure 22(b) presents the efficiency of both techniques over time when cost is fixed (similar to Figure 15(a)) The SnapKS line is derived from comparing full snapshots taken t seconds after the reference snapshot. This figure shows a behavior roughly similar to our simulation result in Figure 18(a). Both techniques exhibit the same efficiency that cross the SnapKS and follows that line. The difference between this result and our simulation is primarily due to the difference in the duration of a walk. More specifically, each step of the walk takes on average around 0.5 second in our simulations (with λ=64 and k=0.59) and 3 seconds in our experiment mainly due to numerous connection timeouts. The longer walk length in the experiment leads to larger changes in the degree distribution and affect the observed behavior. Similar to our simulation results, Figure 22(b) indicates that increasing the cost improves the efficiency of both sampling techniques but as the time elapses its accuracy is limited by the SnapKS line. In summary, while it is difficult to directly compare our results from simulations and experiments, they all demonstrate that both techniques exhibit similar behavior and their efficiency is determined by the cumulative effect of the underlying dynamics on the distribution of desired property.

3.2 Validation over Friendster

Friendster was founded in March 2002 and has more than 65 million users. It sequentially assigns an integer to new users as their ID. We used and validated third party code from TripAdvisor[9] to obtain the information associated with a Friendster user including user profile (which contains name, age, gender, location, and the month that user joined the system among other information), the user's friend list (and their friends' IDs) and a list of photos posted by user[10]. We use the friend

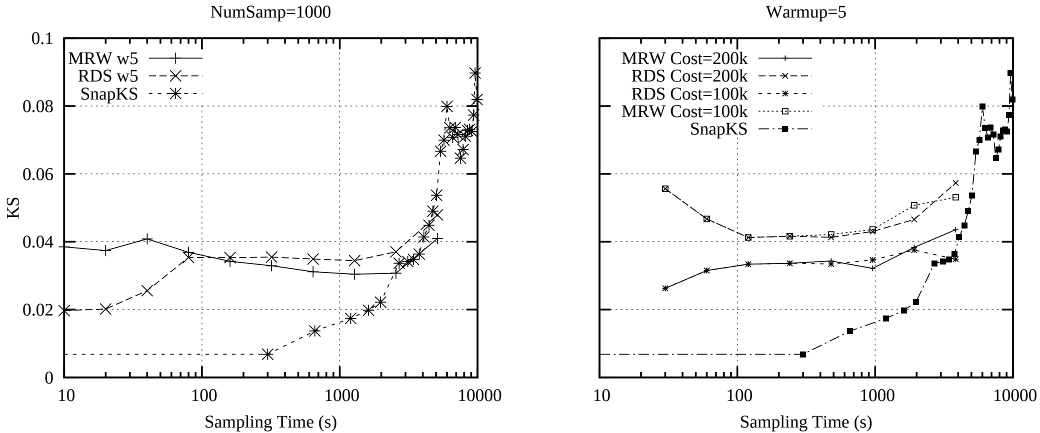
[9] http://code.google.com/p/friendster-java-api/

[10] We register a new "Friendster App" and use the provided API key and the shared secret key to obtain the public in-



(a) DEG (b) RTT (c) UT

Figure 21: Effect of Target node degree on MRW efficiency, $\lambda = 64$, $k = 0.59$, $deg = 30$, NumSamplers=8092, Warmup=6



(a) Fixed Number of Samplers (b) Fixed Sampling Cost

Figure 22: Sampling Gnutella P2P Network

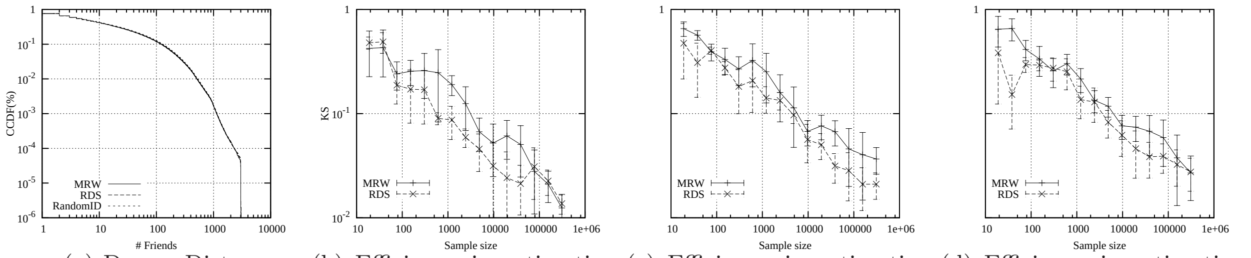
list (and their friends' IDs) to select the next hop for random walks and collect properties of selected users as samples of a desired property.

We primarily focus on three user properties: *(i)* number of friends (or node degree), *(ii)* the length of time that a user has been a member of the system, and *(iii)* user age. The first two properties interact with the walkers and may affect the accuracy of sampling. The third property is added as an independent property for cross validation.

We used 5 parallel MRW walkers and 5 parallel RDS walkers to collect our samples. We also selected a random set of users by generating random IDs and obtaining the information for the associated user. Table 3 summarizes the information for collected samples from each method. The "total" column shows the total number of users, the "private" column indicates the percentage of users whose information were private, and the "singleton" column shows the percentage of visited users that did not list any friends[11]. The "skipped" column presents the percentage of users whose information were obtained but they were not selected as a next step by the corresponding walker. Such skip events occur only for the MRW walkers when the walker determines the probability of going to the next node and ends up staying at the same node. The "useful" column presents the actual number of collected samples by each method that were useful for our analysis after removing private and singleton users. Finally, the "unique" column lists the percentage of useful users that are unique. While both sampling methods use variants of random walk, the percentage of unique users for RDS is significantly higher than MRW since MRW allows the walker to stay in the same node during a step (*i.e.*, whenever it skips going to the next hop) and also it often gets trapped

formation associated with each user.

[11] Singleton users can only be captured by generating random IDs.



(a) Degree Dist.

(b) Efficiency in estimating degree

(c) Efficiency in estimating user age

(d) Efficiency in estimating time since user joined the system

Figure 23: Sampling the online social network, *Friendster*

Table 3: Summary of collected samples from Friendster with two sampling methods and random user selection

Tech.	Start Date, Time	Total users	Useful(%)	Private(%)	Singletons(%)	Skipped(%)	Unique(%)
MRW	04/26/08, 14:20	3.4M	49.0	16.0	-	35.0	21.7
RDS	04/24/08, 18:22	3.2M	65.0	35.0	-	-	92.7
Random	04/24/08, 17:00	604K	55.6	12.4	31.9	-	99.5

in low degree clusters. The average rate of progress by each walker was 71.7 users/min that translates to 110.0 queries/min [12].

Figure 23 depicts several characteristics of both sampling methods over Friendster based on the above three datasets. The results suggest that the percentage of Friendster users with degree higher than 3000 is very low.

Figure 23(a) presents the CCDF of node degree based on samples collected by the RDS and MRW methods as well as randomly selected users on a log-log scale. This figure reveals that the distribution of node degree from all three datasets is very close ($KS \leq 0.0125$). Figure 23(b) presents the average KS error in degree distribution for each sampling technique, with respect to the distribution derived from randomly selected users (as a reference), as a function of number of samples for both RDS and MRW. The variability of the KS value for samples from different walkers are shown with vertical bars. Figure 23(c) shows the accuracy of the samples collected by both methods in estimating the distribution of each user's age. Both figures indicate that the quality of samples rapidly improves with the sample size. The RDS method exhibits better accuracy in most scenarios. This can be attributed to the ability of RDS to collect unique samples and is in agreement with the findings from our simulations presented earlier.

4. REFERENCES

[1] A.-L. Barabasi, Z. Dezso, E. Ravasz, S.-H. Yook, and Z. Oltvai. Scale-free and Hierarchical Structures in Complex Networks. In *Seventh Granada Lectures*, 2002.

[2] B. Bollobás. A probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

[3] H. Dämpfling. Gnutella Web Caching System: Version 2 Specifications Client Developers' Guide. http://www.gnucleus.com/gwebcache/newgwc.html, June 2003.

[4] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *Internet Measurement Workshop*, Nov. 2002.

[5] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *IMC*, Oct. 2007.

[6] D. Stutzbach and R. Rejaie. Understanding Churn in Peer-to-Peer Networks. In *Internet Measurement Conference*, Oct. 2006.

[7] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On Unbiased Sampling for Unstructured Peer-to-Peer Networks. In *IMC*, Oct. 2006.

[8] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On Unbiased Sampling for Unstructured Peer-to-Peer Networks. *TON*, 2008.

[9] D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. In *Internet Measurement Conference*, pages 49–62, Oct. 2005.

[10] D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. *TON*, 16(2), Apr. 2008.

[11] D. J. Watts. Six Degrees. In *The Essence of a Connected Edge*. ACM Press, 2003.

[12] Our data collection was stopped since Friendster disabled our application key after a few days.