Empirical Characterization of P2P Systems

Reza Rejaie
Mirage Research Group
Department of Computer & Information Science
University of Oregon
http://mirage.cs.uoregon.edu/

Collaborators: Daniel Stutzbach (Univ. of Oregon), Nick Duffield, Shubho Sen, Walter Willinger (AT&T Research)

Supported by NSF, Cisco, Thomson Labs

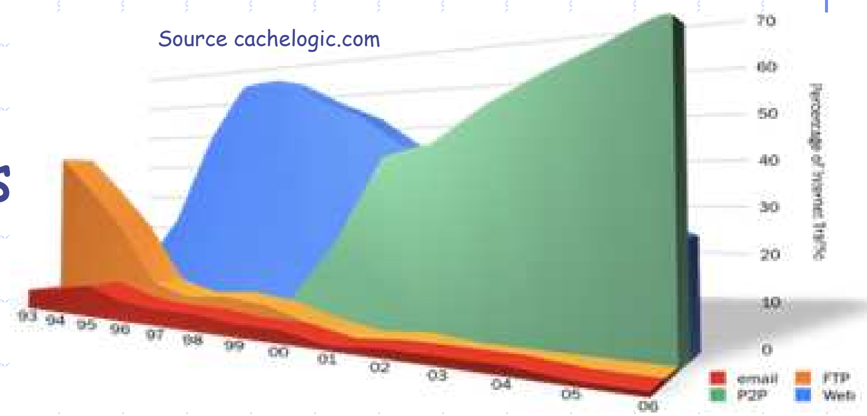
Introduction

- P2P applications are very popular over the Internet
  - File-sharing: Gnutella, Kazza, eDonkey
  - Content distribution: BitTorrent
  - IP telephony: Skype
- P2P applications remain popular because of
  - Ease of deployment, self-scaling, infrastructure-less
- Significant impact on the Internet
- Characterizing P2P applications is essential for
  - Evaluating their performance and improving their designs
  - Conducting meaningful simulations and analytical study
  - Examining their impact on the network

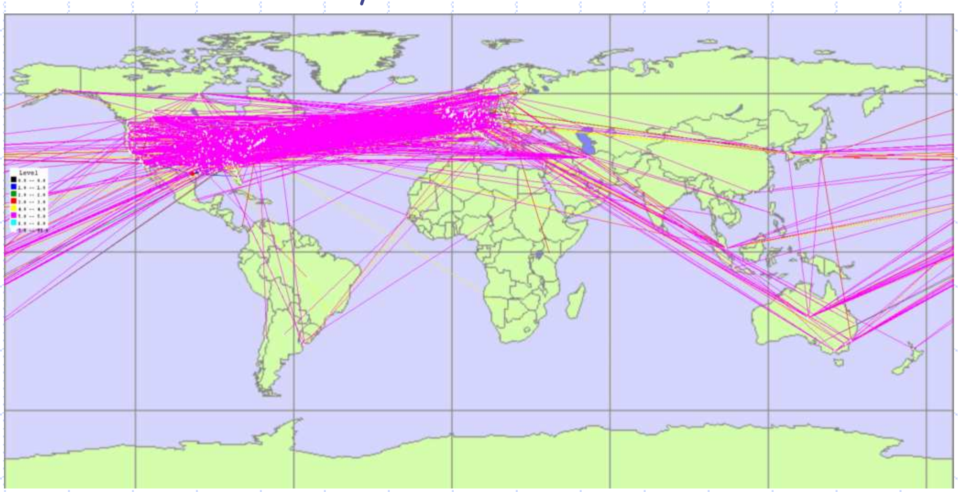
➢ *Characterizing large scale P2P applications is very difficult!*

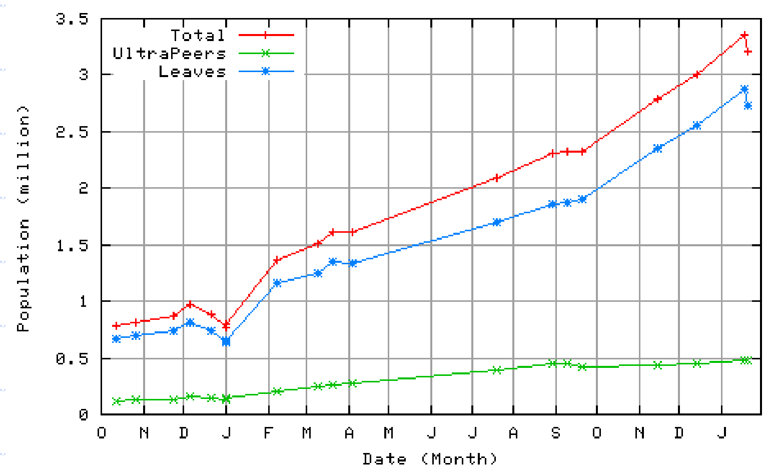
Effect on the Internet

- 70% of all Internet traffic
  [CacheLogic Research 2006]
- Some P2P apps have millions of simultaneous users.
- Geographically distributed.



Gnutella overlay

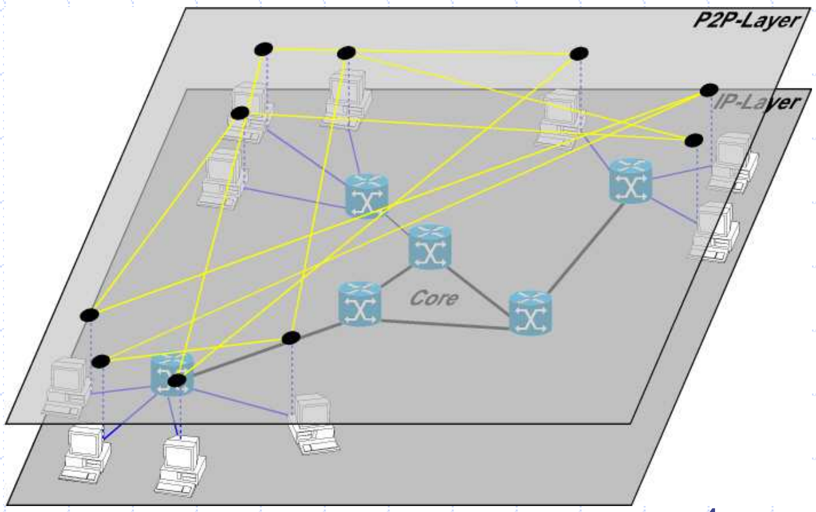




Gnutella population (Oct 04 – Jan 06)

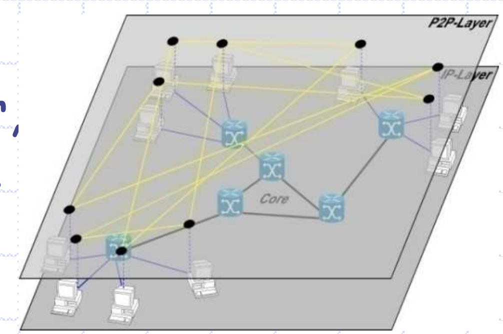
P2P Systems: An Overview

- Theme: a group of peers (end-systems) connect together to share their resources
  - e.g. bandwidth, CPU, storage
  - No special support from the network is needed
- As participating peers arbitrarily join & leave, they form an (application level) overlay topology.
  - Overlay is inherently dynamic
- Two flavors:
  - Structured (DHT)
  - Unstructured



Empirical study of P2P Systems

- Characterizing P2P applications requires capturing "snapshots" of the systems.
  - Snapshot is a graph that represents state of the system at a given point of time (peers = nodes, connections = edges).
  - Individual snapshots reveal **instantaneous properties**.
  - Consecutive snapshots reveal **dynamics**.
- Ideally, a snapshot is captured instantaneously.
- In practice, a snapshot is progressively discovered by a P2P crawler.
  - P2P apps should provide support for crawler, e.g. query a peer for list of neighbors, files.
  - ➢ It is difficult to characterize proprietary P2P applications.



The Problem

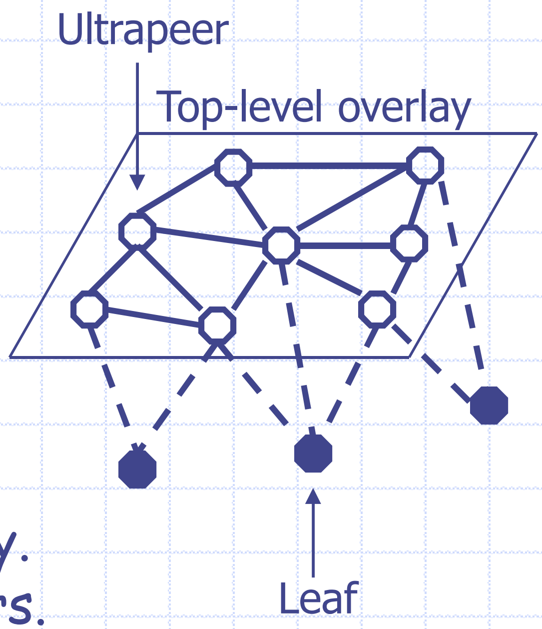
- The overlay is large & rapidly changing during a crawl
  - ***Captured snapshots are likely to be "distorted."***
  - *Increasing crawler speed 1) reduces distortion in captured snapshots, and 2) improves granularity of captured dynamics.*
- Previous empirical studies captured either
  - Complete snapshots with slow crawlers => distorted, or
  - Partial snapshots => less distorted, may not be representative
  - Accuracy of captured snapshots have not been examined.
  - Primary focus on the analysis of snapshots
- Our approach:
  - Capturing **complete snapshots** with a fast crawler
  - Capturing **representative samples** of the system

The IonP2P Project

- Capturing accurate & complete snapshots
  - Cruiser: a fast P2P crawler [GI 05]
- Several empirical characterizations
  1) Unstructured Overlay Topology [IMC 05]
  2) Structured Overlay Topology [INFOCOM 06]
  3) Churn [IMC 06]
  4) Available files [MSJ 07]
- Unbiased Sampling of large and dynamic P2P Systems [IMC 06]

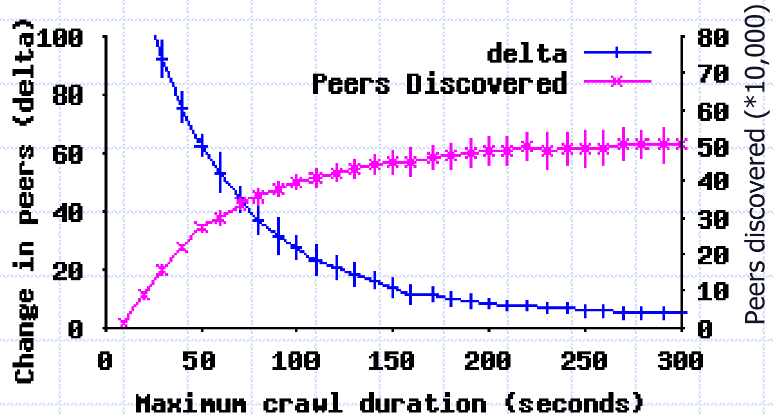
Characterizing Unstructured Overlay

- Focus on Gnutella because of its …
  - Large size (1M+ peers)
  - Mature implementations
  - Open specifications
- Gnutella uses a two-tier overlay.
  - Improves scalability.
  - Ultrapeers form an unstructured overlay.
  - Leaf peers connect to multiple ultrapeers.
  - eDonkey, FastTrack are similar
- Goal: to characterize ***graph properties*** & ***dynamics*** of the top-level unstructured overlay topology



Evaluating Snapshot Accuracy

- Developed a fast crawler, Cruiser
- No reference snapshot to compare
- Completeness of snapshots: fraction of edges, nodes captured
- Tradeoff between granularity & completeness of snapshots
  - Node distortion > 8%
  - Edge distortion > 15%



Data Set

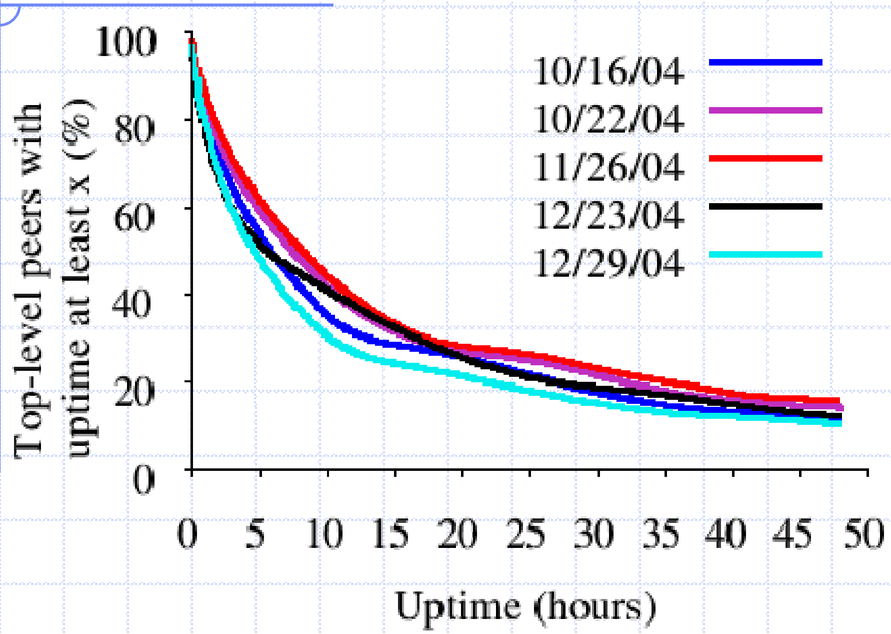
- 100K+ Gnutella snapshots, over the past 4 years
- To examine static properties, we focus on four:

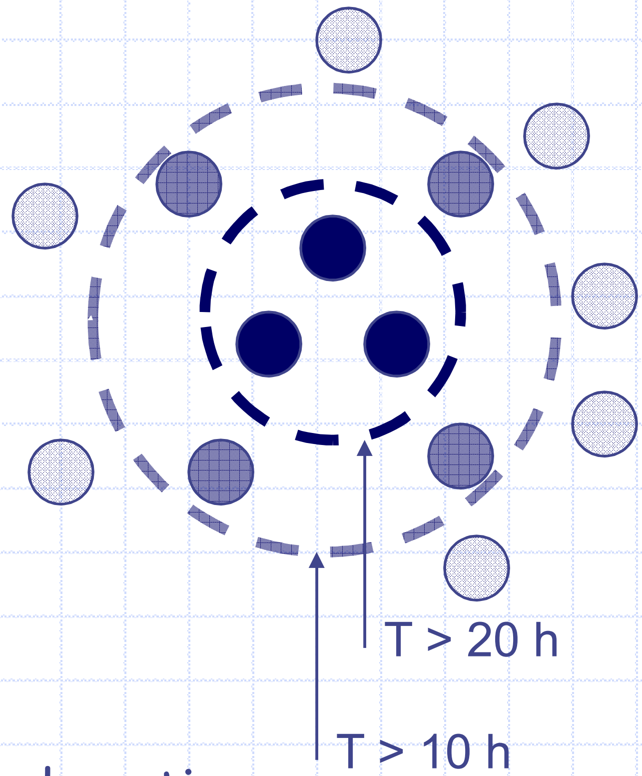
Date	Total Nodes	Leaves	Ultrapeers	Top-level Edges
9/27/04	725,120	614,912	110,208	1,212,772
10/11/04	779,535	662,568	116,967	1,244,219
10/18/04	806,948	686,719	120,229	1,331,745
2/2/05	1,031,471	873,130	158,345	1,964,121

- To examine dynamic properties, we use slices:
  - Each slice is 2 days of ~500 back-to-back snapshots
  - Captured starting 10/14/04, 10/21/04, 11/25/04, 12/21/04, and 12/27/04

Characterizing Unstructured Overlay/

Stable Core



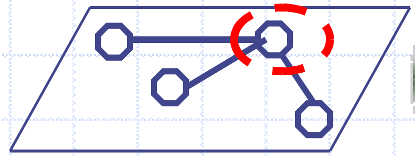


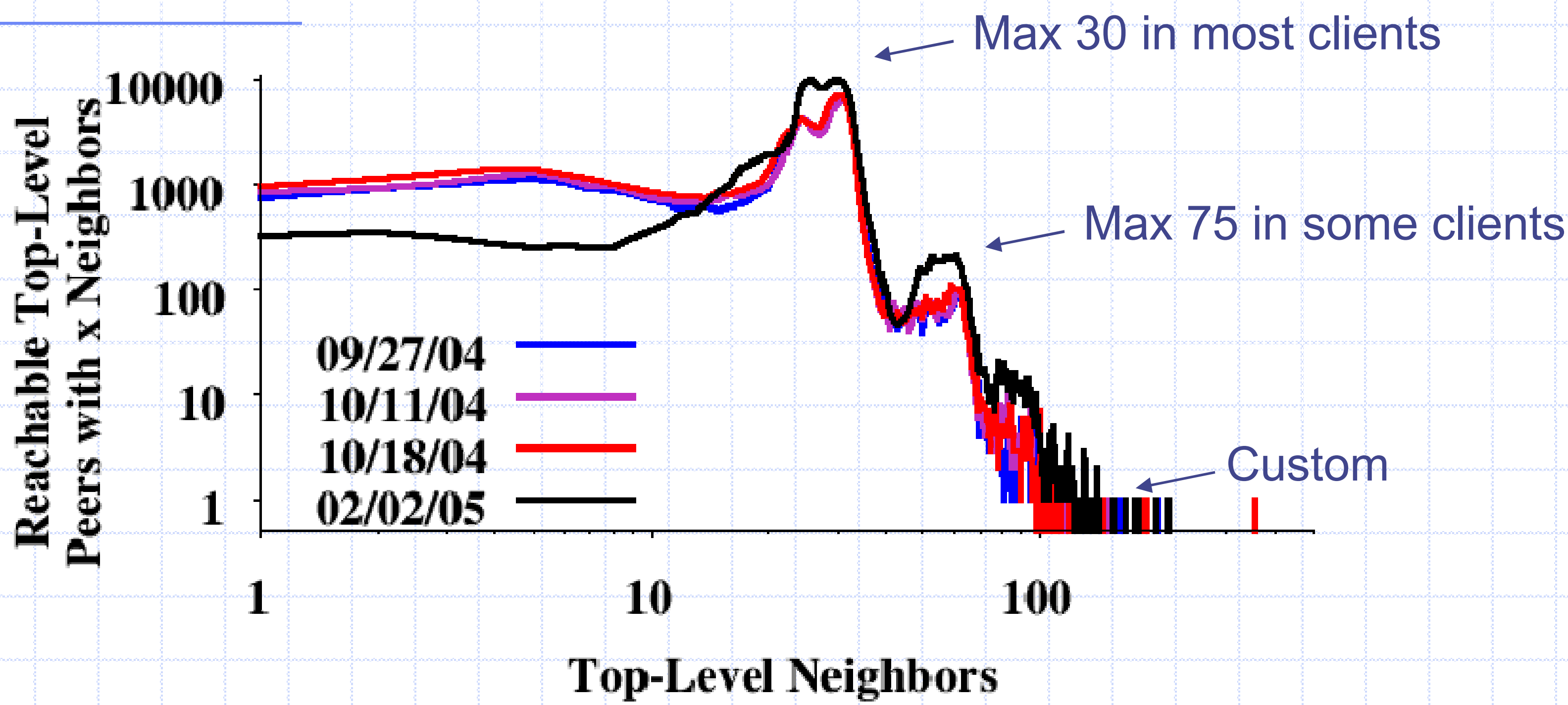
- Most peers have a short uptime.
- Other peers have been around for a long time.
- **Stable core**: a set of peers with uptime higher than a threshold (T).
  - Higher threshold => more stable group of peers

Summary of Characterizations

- Graph Properties
  - Implementation heterogeneity
  - Degree Distribution:
    - **Top-level degree distribution**
    - Ultrapeer-leaf connectivity
    - Degree-distance correlation
  - Reachability:
    - Path lengths
    - Eccentricity
  - Small world properties
  - Resiliency to peer departure
- Dynamic Properties
  - Existence of stable core:
    - Uptime distribution
    - Biased connectivity
  - Properties of stable core:
    - Largest connected component
    - Path lengths
    - Clustering coefficient

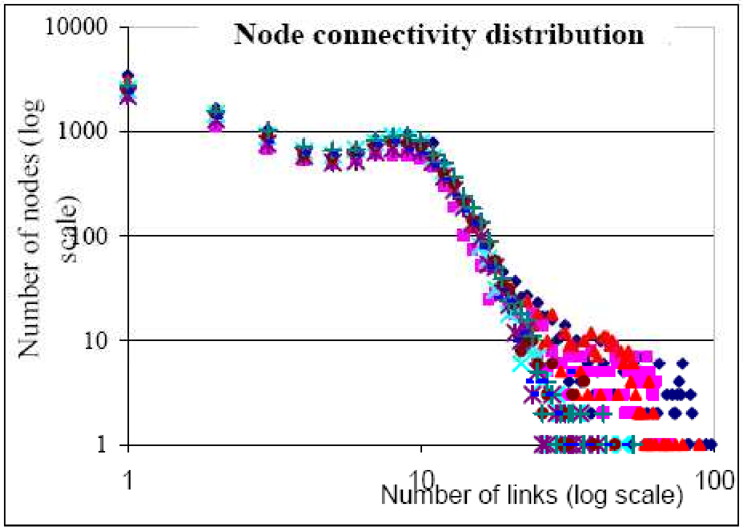
Top-level Degree

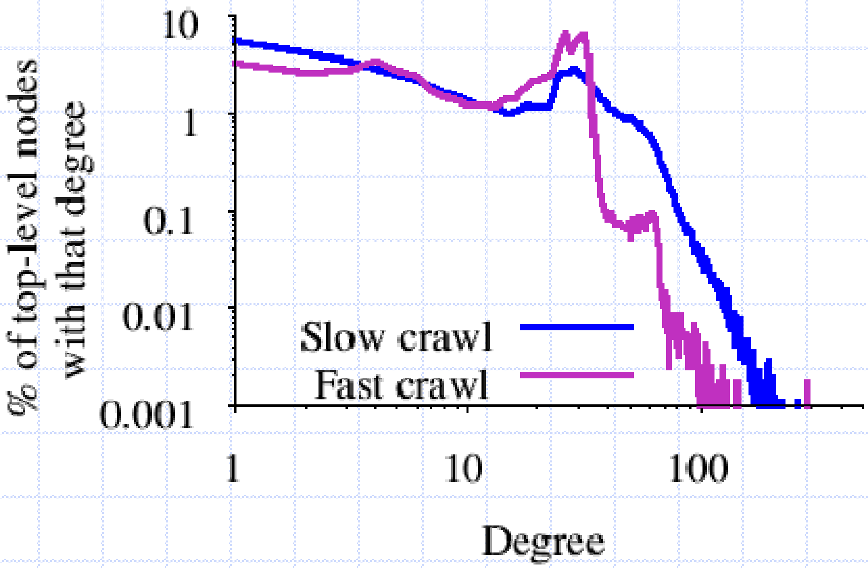




- This is the degree distribution among ultrapeers in Gnutella.
- There are obvious peaks at 30 and 75 neighbors.
- A substantial number of ultrapeers have fewer than 30.
- ***What happened to the power-law reported by prior studies?***

What happened to power-law?





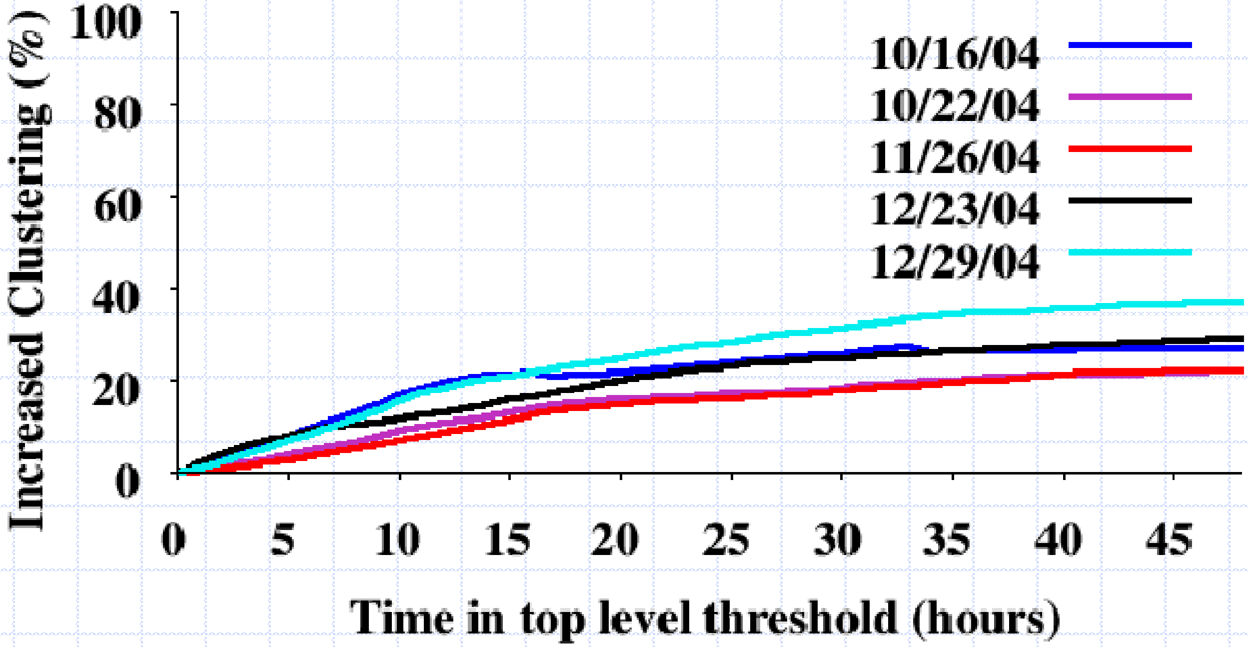
[Ripeanu 02 ICJ]

- When a crawl is slow, many short-lived peers report long-lived peers as neighbors.
- But those neighbors are not all present **at the same time**.
- Degree distribution from a slow crawl resembles prior results.

Biased Connectivity

- Hypothesis: long-lived nodes tend to be more connected to other long-lived nodes
  - Rationale: Once connected, they stay connected.
  - Long-lived peers have more opportunities to become neighbor.
- To quantify bias in the connectivity of the stable core:
  - Randomize the edges to create a graph **without** biased connectivity.
  - Ratio of the edges in the observed stable core with the comparable randomized graph.

Stable Core Edges



- 20%—40% more edges in the stable core compared to random.
- Connectivity exhibits an **onion-like biased** connectivity where peers are more likely to connect to other peers with same/higher uptime. (other properties in the paper)
- ***Despite high churn, there is a relatively stable "backbone".***

Why do we need to sample?

- Capturing an accurate & complete snapshot is hard and might even be infeasible for large systems
  - P2P systems are distributed, large, and rapidly changing.
  - Capturing a global picture is time-consuming, resulting in a blurry picture.
- Sampling is a natural approach, and has been *implicitly* used in earlier P2P measurement studies.
- *But how do we know the samples are representative?*

Sampling Unstructured P2P Networks

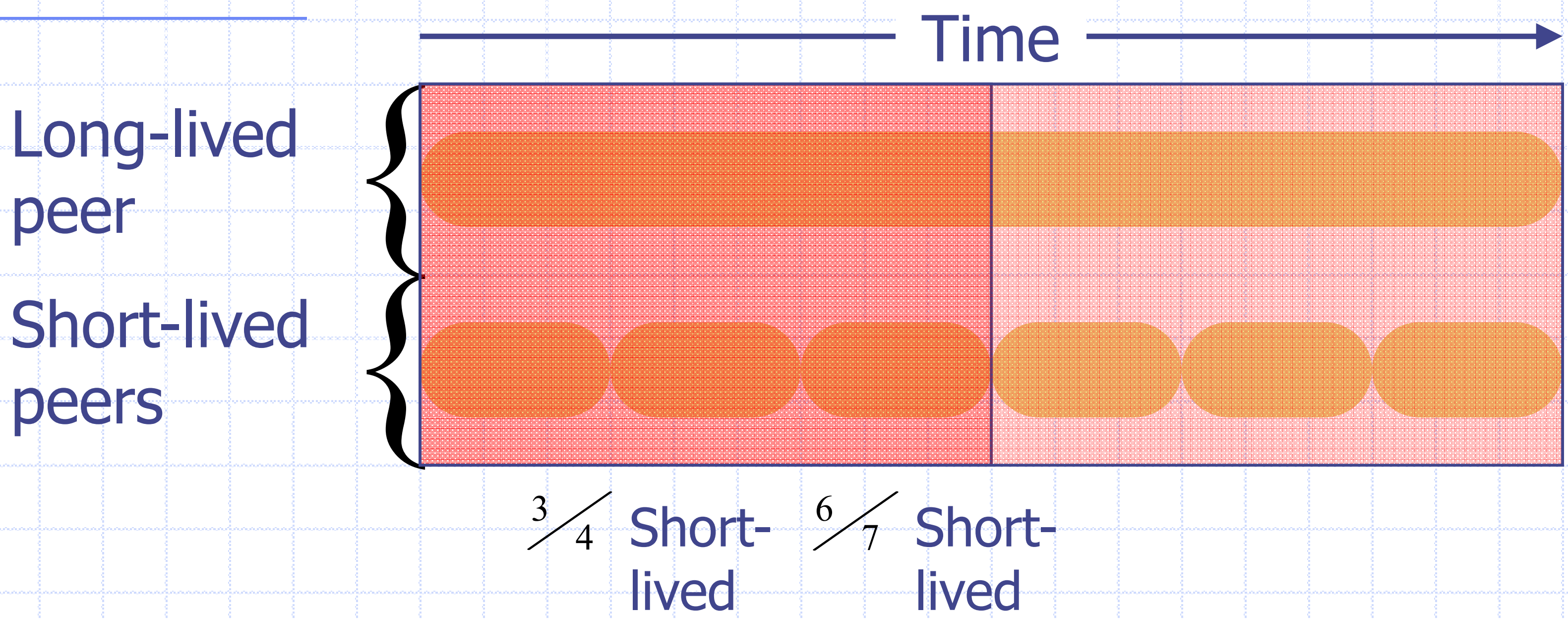
- We focus on sampling *peer properties.*
  - Number of neighbors (degree)
  - Link bandwidth
  - Number of shared files
  - Remaining uptime
- Sampling peer properties occurs in two steps:
  - Discover and select peers
  - Collect measurements from the selected peers
- Selecting peers *uniformly at random* is hard.
  - **Temporal:** Peer dynamics can introduce bias.
  - **Topological:** The graph topology can introduce bias.
  - First, we examine these two problems in isolation.
  - Then, we examine both problems together.

Sampling with Dynamics

- Define V_t as the set of peers present at time t.
- We gather samples over a measurement window of length Δ.
- The most common approach is to gather peers from the set present during the window:

$$v \in \bigcup_{t=t_0}^{t_0+\Delta} V_t$$

Bias towards Short-Lived Peers



- Consider a simple two-tier, containing:
  - One long-lived peer
  - One rapidly-changing short-lived peer
- The common approach over-selects short-lived peers

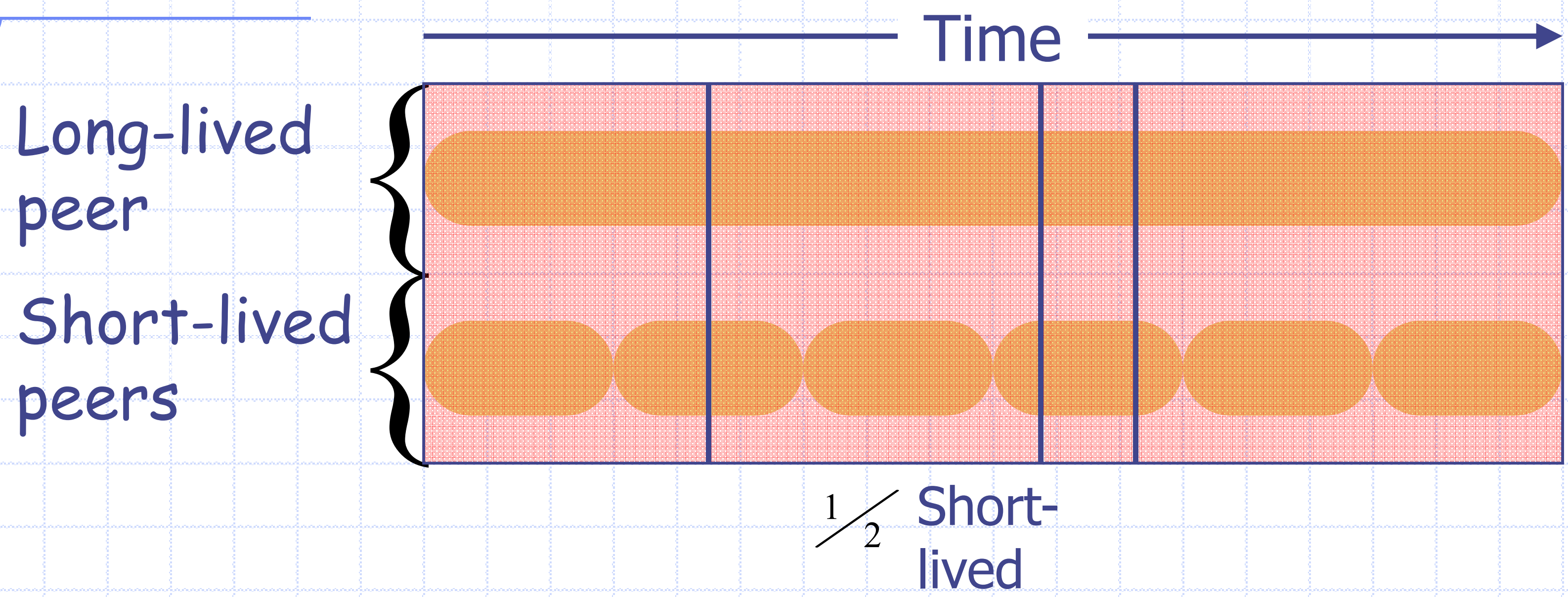
Handling Temporal Causes of Bias

$$\cancel{v_i \in \bigcup_{t=t_0}^{t_0+\Delta} V_t}$$

- The common approach is intuitive but incorrect.
- Sampling peers is the wrong goal.
- We want to sample *peer properties*.
- Two samples from the same peer, but at different times, are distinct.
- Allow sampling the same peer more than once, at different points in time.

$$t \in [t_0, t_0 + \Delta] \quad v_{i,t} \in V_t$$

Example of avoiding bias



- Allowing re-selecting a peer solves the problem.
- The long-lived peer will be selected half the time, reflecting the actual state of the system.
- How do we select a peer uniformly at random at a particular moment?

Sampling from Static Graphs

- Assume for the moment a static graph…
- Goal: Select a peer uniformly from the graph
- Discover:
  - Begin with one peer.
  - Query peers to discover neighbors.
  - Classic algorithms: Breadth-First Search, Depth-First Search
- Select:
  - Choose a subset of discovered peers
  - Gather samples from the selected peers

Advantages of Random Walks

- Problems with classic approaches:
  - Peers are correlated by their neighbor relationship
  - Peers with higher degree discovered more often
  - A peer can only be selected once.

- Random walks are a promising alternative:
  - The information in the starting location is "lost" by repeatedly injecting randomness at each step.
  - The results are biased, but the bias is precisely known.
  - Random walks can implicitly visit the same peer twice.

Random walks, formally

- Random walks can be described with a transition matrix, *P(x,y)*.
- *P(x,y)* is the probability of moving from *x* to *y*.

$$P(x,y) = \begin{cases} \dfrac{1}{\deg(x)} & y \text{ is a neighbor of } x \\ 0 & \text{otherwise} \end{cases}$$

- $P^r(x,y)$ is the probability of moving from *x* to *y* after *r* moves
- Random walks converge to a stationary distribution:

$$\pi(x) = \lim_{r\to\infty} (\nu P^r)(x) = \frac{\deg(x)}{2|E|}$$

- Problem: we want a uniform distribution:

$$\mu(x) = \frac{1}{|V|}$$

The Metropolis—Hastings Method

- The Metropolis—Hastings method modifies the transition matrix to yield the desired distribution:

$$Q(x,y)=\begin{cases} P(x,y)\min\left(\frac{\mu(y)P(y,x)}{\mu(x)P(x,y)},1\right) & \text{if } x\neq y \\ 1-\sum_{x\neq y} Q(x,y) & \text{if } x=y \end{cases}$$

- Proven for static graphs
- Plugging in our *P(x,y)* and *μ(x)*:
  - Select a neighbor *y* of *x* uniformly at random
  - Transition to *y* with probability deg(*x*) / deg(*y*)
  - Otherwise, self-transition to *x*.

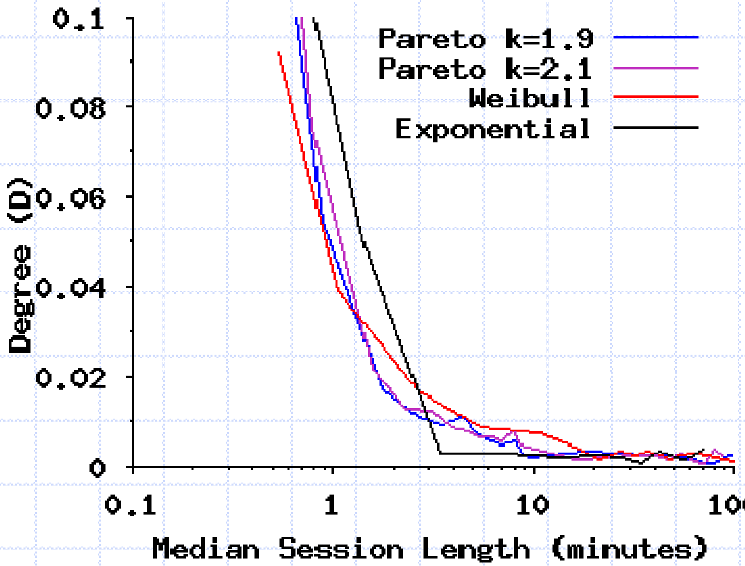
Sampling from Dynamic Graphs

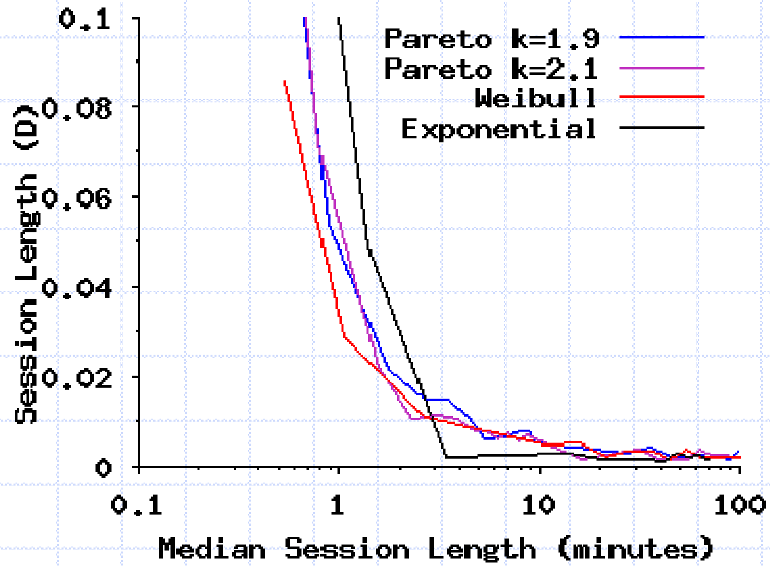
- Coping with departing peers
  - We maintain a stack of visited peers
  - If a query times out, go back in the stack
- Hypothesis: A Metropolized random walk will yield approximately unbiased samples in practice.
  - Trivially valid for extremely slowly changing graphs
  - Trivially false for extremely rapidly changing graphs
  - Where is the transition?
- Methodology:
  - Session-level simulations of a wide variety of situations
  - Determine what conditions lead to biased samples
  - Do those conditions arise in practice?

Metrics: Fundamental properties

- We focus on three fundamental properties that affect the walk:
  - Degree
  - Session length
  - Query latency
  - Use the Kolmogorov-Smirnov (KS) statistic (D) for each distribution versus a snapshot from an oracle.
- We evaluate these metrics under a variety of conditions:
  - Several models of churn
  - Several models of degree distribution
  - Four different peer discovery mechanisms

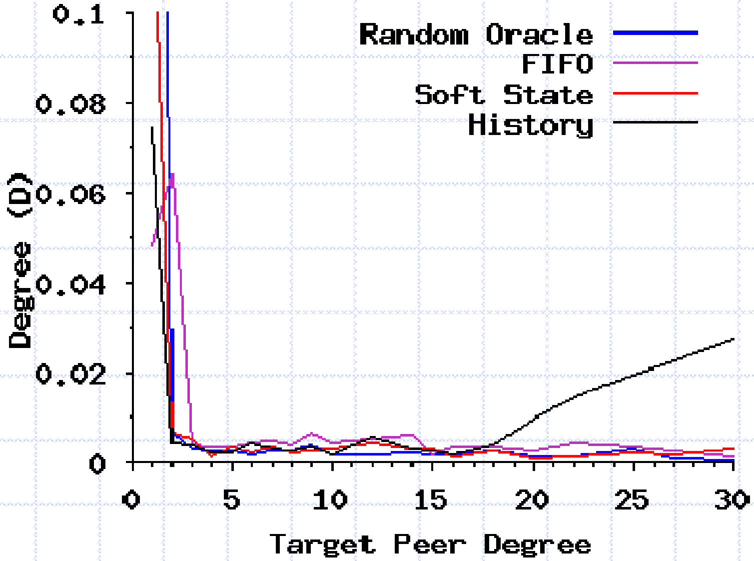
Varying churn

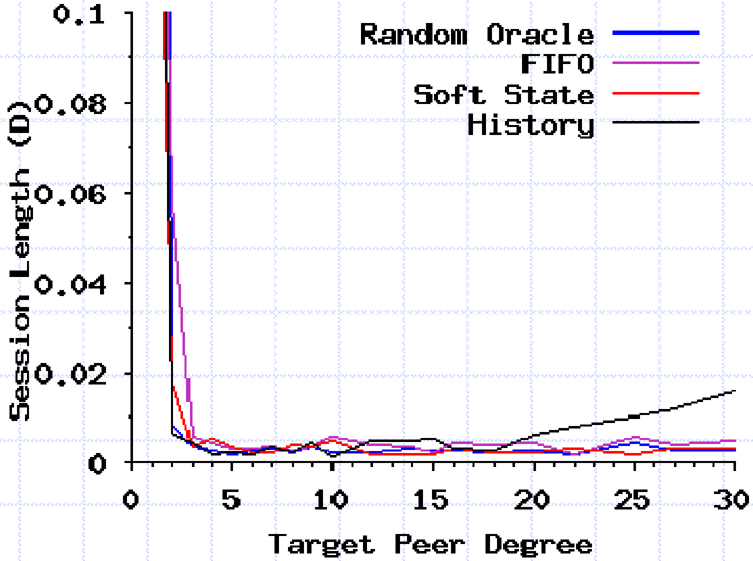




- Each point represents a simulation; y-axis shows KS statistic (D)
- Error is low over a wide range of session lengths
- Becomes significant for median < 2 min
- High for median < 30 s
- Type of distribution does not have a large impact

Varying topology





- Little bias when target degree > 2
- Degree ≤ 2 means network fragmentation
- History mechanism bias is due to ~2% of peers with no neighbors.

Conclusions

- Characterizing P2P systems is very challenging
  - Capturing accurate snapshot is hard
- Using rather accurate snapshots, we characterize key properties of P2P systems
  - Debunked some of the commonly reported results
- Sampling is a promising approach
  - But temporal & topological bias can lead to bias.
- Metropolized Random Walk with Backtracking (MRWB) technique provides unbiased samples.
- Unique resources for researchers and practitioners
  - archive of P2P snapshots (1.5 TB), measurement tools, models, etc

Selected Publications

- Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems, **Transactions on Networking** 2007
- Characterizing Files in the Modern Gnutella Network, **Multimedia Systems Journal** 2007
- On Unbiased Sampling for Unstructured Peer-to-Peer Networks, **Internet Measurement Conference** 2006
- Understanding Churn in Peer-to-Peer Networks, **Internet Measurement Conference** 2006
- Understanding Peer-Level Performance in BitTorrent: A Measurement Study, **ICCCN 2007** (chair recommended paper)
- On the Long-term Evolution of the Two-tier Gnutella Overlay, **Global Internet** 2006
- Improving Lookup Performance over a Widely-Deployed DHT, **INFOCOM** 2007

Visit http://mirage.cs.uoregon.edu/P2P for more information

Online Social Networks

- Users are first-class objects in OSNs
  - MySpace, Flickr, Facebook, YouTube, …
- Users provide their friend lists
  - Friendship relation could be bi- or unidirectional
- All users collectively form a ***friendship graph***
- Friendship graph can be used to crawl OSNs
- Friendship graph is evolving over time
- OSNs often limit the rate of crawling the system (e.g. friendship graph)
  - *Sampling seems to be a promising approach*

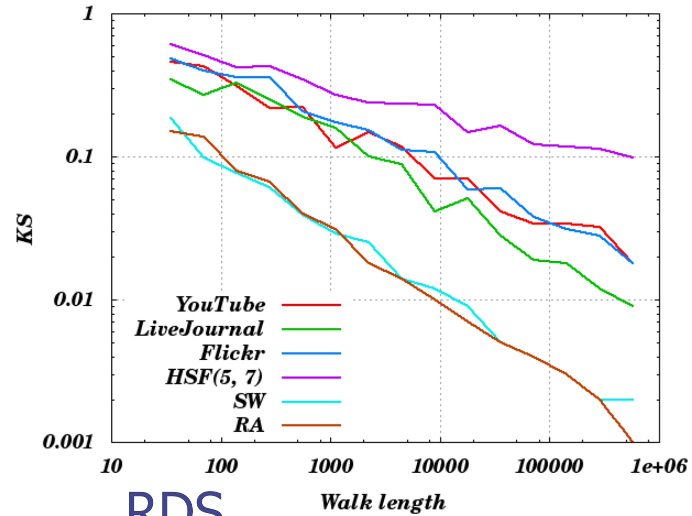
Sampling Online Social Nets

- The abstract graph sampling problem in OSNs is similar to P2P networks
- New challenges:
  - Friendship graph is directed in some OSNs, a walker is trapped in dead-end regions
  - Friendship graph exhibits different clustering properties
  - The system may not provide all friends of a popular user (e.g. YouTube)!
  - System API may change during a crawl !

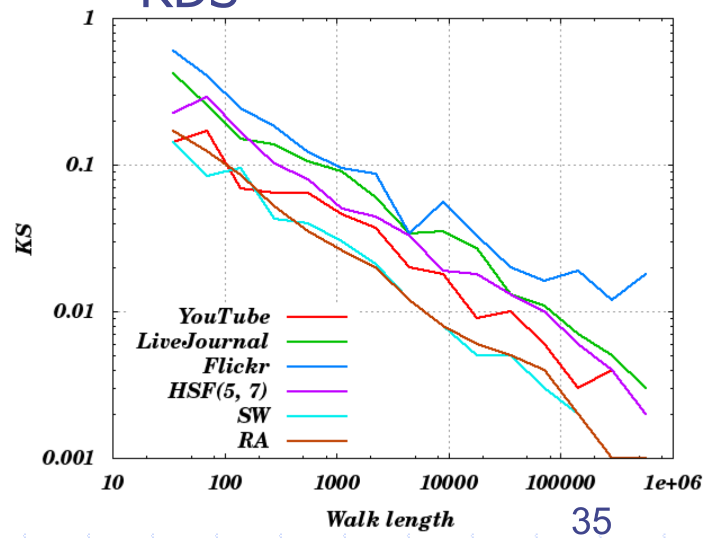
Preliminary Results

- Focusing on undirected OSNs
- MRW and RDS techniques exhibit lower efficiency over OSNs:
  - Unknown interactions between random walk and graph structure
- OSN-like graphs consist of many small, low-degree clusters that are connected through high-degree nodes
- *Probability of sampling nodes in a cluster depends on its incoming edges (not node degree)!!*

MRW



RDS



Challenges in Sampling OSNs

- Measurement Techniques
  - Coping with clustering properties of OSN graphs
  - Unbiased sampling in a directed graphs
- Characterizing OSNs
  - the evolution of OSN friendship graph
  - the correlation between the friendship and interaction graphs
  - Identifying underlying causes