Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers

Daniel Stutzbach and Reza Rejaie

University of Oregon, Eugene OR 97403, USA
{agthorr, reza}@cs.uoregon.edu

Abstract. The increasing popularity of Peer-to-Peer (P2P) networks has led to growing interest in characterizing their topology and dynamics [1, 2, 3, 4], essential for proper design and effective evaluation. A common technique is to capture topology snapshots using a crawler. However, previous studies have not verified the accuracy of their captured snapshots. We present techniques to measure the inaccuracy of topology snapshots, quantify the effects of unreachable peers and crawling speed, and explore the impact of snapshot accuracy on derived characterizations.

1 Introduction

The accuracy of captured snapshots by P2P crawlers can be significantly affected by both the duration of a crawl and the ratio of unreachable peers. Determining the accuracy of captured snapshots of a P2P system is fundamentally difficult because a perfect reference snapshot for comparison is not available. The desired characterization of P2P systems determines the granularity and type of collected information in each snapshot, in the form of a tradeoff between the duration of a crawl and the completeness of the captured snapshot. For example, studying churn only requires a list of participating peers, and a crawler can gather this information from a subset of all peers with reasonable accuracy. In contrast, to study the overlay topology a captured snapshot should include all edges of the overlay; this requires the crawler to directly contact every peer, otherwise a connection between two unvisited peers would be missed.

To study snapshot accuracy, we developed a fast and efficient Gnutella crawler, called *Cruiser*, that is able to capture a complete snapshot of the Gnutella network in around 5 minutes with six off-the-shelf desktop PCs. Previous studies typically crawled their target P2P systems in 30 minutes to two hours (e.g., [5, 4]), despite crawling significantly smaller networks. Cruiser achieves this significant reduction in crawl time as follows: *(i)* it leverages several features of modern Gnutella, including its semi-structured topology and efficient new handshake mechanism; *(ii)* it substantially increases the degree of concurrency during the crawling process by deploying a master-slave architecture and allowing each slave crawler to contact hundreds of peers simultaneously. More details on the design and evaluation of Cruiser may be found in our tech report [6].

C. Dovrolis (Ed.): PAM 2005, LNCS 3431, pp. 356–360, 2005.

2 Modern Gnutella

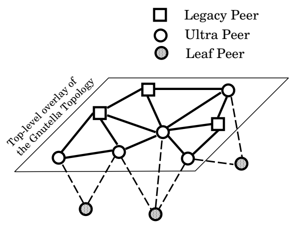


Fig. 1. Semi-Structured Topology of Modern Gnutella

We briefly describe the key features of modern Gnutella [7, 8] that are used by Cruiser. The original Gnutella protocol had limited scalability due to its flat overlay. To address this limitation, most modern Gnutella clients implement a two-tiered network structure by dividing peers into two groups: *ultrapeers* and *leaf* peers. As shown in Fig. 1, each ultrapeer neighbors with several other ultrapeers within a top-level overlay. The majority of the peers are leaves that are connected to the overlay through a few ultrapeers. Those peers that do not implement the ultrapeer feature can only reside in the top-level overlay and do not accept any leaves. We refer to these peers as *legacy* peers. We also refer to the legacy peers and ultrapeers collectively as the *top-level* peers.

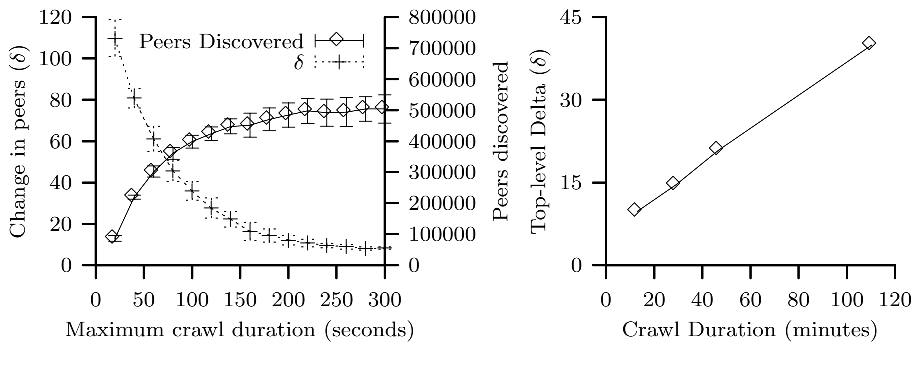
Also, modern Gnutella clients implement a special handshaking feature that enables the crawler to quickly query a peer for a list of its current neighbors. Previous crawlers relied on other features of the Gnutella protocol, namely Ping-Pong messages, to retrieve this information, but these techniques were less efficient.

3 Accuracy of Captured Snapshots

We consider three effects that can impact the accuracy of topology snapshots. First, we consider unreachable peers which, for one reason or another, cannot be crawled. Second, we consider how much accuracy can be maintained while cutting short the duration of crawls. Finally, we consider the impact of the crawler's speed.

Unreachable Peers: A non-negligible subset of contacted peers in each crawl time out (15–24%), prematurely drop (6–10%) or refuse TCP connections (5–7%). Peers are unreachable when they have already left the system (i.e., departed), they are located behind a firewall (or NATed), or they receive SYN packets at too high a rate (i.e., overloaded). Departed and firewalled peers are noted in previous studies; however we find many unreachable peers are overloaded, refusing and accepting TCP connections sporadically over a short period of time (i.e., within a single minute they alternate repeatedly). Unreachable ultrapeers can introduce the following errors in a captured snapshot: *(i)* including departed peers, *(ii)* omitting branches between unreachable ultrapeers and their leaves, and *(iii)* omitting branches between two unreachable top-level peers. To minimize these errors, it is important to quantify what portion of unreachable peers were departed versus firewalled or overloaded. Unfortunately, there is no reliable test to firmly verify the status of unreachable peers among the three possible scenarios, since overloaded, firewalled, and departed peers may or may not reply to SYN packets. However, we found that repeatedly attempting to connect to peers which have timed out is unlikely to ever meet with success, even after attempting for several hours. This suggests that those peers, at least, are firewalled.

Impact of Crawling Duration: To examine the impact of crawl duration on the accuracy of captured snapshots, we modified Cruiser to stop the crawl after a specified period. Shorter crawls allow us to capture back-to-back snapshots more rapidly, which increases the granularity for studying churn. We performed two back-to-back crawls and repeated this process for different durations. We define δ_+ and δ_- as the number of new and missing peers in the second snapshot compared to the first one, respectively (normalized by the total number of peers in the first crawl). Figure 2(a) presents the sum $\delta = \delta_+ + \delta_-$ as well as the total number of discovered peers as a function of the crawl duration. During short crawls (the left side of the graph), δ is high because the captured snapshot is incomplete, and each crawl captures a different subset. As the duration of the crawl increases, δ decreases, indicating that the captured snapshot becomes more complete. Increasing the crawl length beyond four minutes does not decrease δ any further, and achieves a marginal increase in number of discovered peers. This figure reveals a few important points. First, there exists a "sweet spot" for the crawl duration beyond which crawling has diminishing returns if the goal is simply to capture the population. Second, the change of $\delta = 0.08$ is an upper-bound on the distortion due to the passage of time as Cruiser runs. Third, for sufficiently long crawls, Cruiser can capture a relatively accurate snapshot. The relatively flat values of delta for longer crawls suggest that a small but significant fraction of the network is unstable and turns over quickly. For shorter durations, the standard deviation of the peers discovered is small, since the size of the discovered topology is limited by the crawl's duration. For longer



(a) Error as a function of maximum crawl duration

(b) Error as a function of crawling speed

Fig. 2. Effects of crawl speed and duration, generated by running two crawls back-to-back per x-value

durations, the standard deviation is larger and measures the actual variations in network size.

Impact of Crawling Speed: To examine the impact of crawling speed on the accuracy of captured snapshots, we decreased the speed of Cruiser by reducing the number of parallel connections that each slave process can open. Figure 2(b) depicts the error in between snapshots from back-to-back crawls as a function of crawl duration. The first snapshot was captured with the maximum speed and serves as a reference, whereas the speed (and thus duration) of the second snapshot has changed. The duration of the second snapshot is shown as the x value. This figure clearly demonstrates that the accuracy of snapshots decreases significantly for longer crawls.

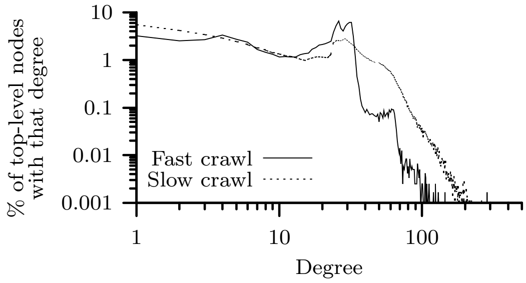


Fig. 3. Observed top-level degree distributions in a slow and a fast crawl

Impact of Snapshot Accuracy on Derived Characterization: To show the effect this error has on conclusions, in Fig. 3 we show the observed degree distribution of a fast crawl versus a crawl limited to 60 concurrent connections. The slow crawl distribution looks similar to that seen in [4][1], which lead to the conclusion that Gnutella has a two-piece power-law degree distribution. If we further limit the speed, the distribution begins to look like a single-piece power-law, the result reported by earlier studies [9, 5]. To a slow crawler, peers with long uptimes appear as high degree because many short-lived peers report them as neighbors. However, this is a misrepresentation since these short-lived peers are not all present at the same time.

4 Conclusion

In this extended abstract, we have developed techniques for examining the accuracy of topology snapshots captured by peer-to-peer crawlers, including demonstrating that earlier conclusions may be incorrect and based on measurement artifacts.

References

1. Bhagwan, R., Savage, S., Voelker, G.: Understanding Availability. In: International Workshop on Peer-to-Peer Systems. (2003)
2. Saroiu, S., Gummadi, P.K., Gribble, S.D.: Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. Multimedia Systems Journal **8** (2002)

[1] Their crawler was limited to 50 concurrent connections.

3. Liben-Nowell, D., Balakrishnan, H., Karger, D.: Analysis of the Evolution of Peer-to-Peer Systems. In: Principles of Distributed Computing, Monterey, CA (2002)
4. Ripeanu, M., Foster, I., Iamnitchi, A.: Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. IEEE Internet Computing Journal **6** (2002)
5. clip2.com: Gnutella: To the Bandwidth Barrier and Beyond (2000)
6. Stutzbach, D., Rejaie, R.: Characterizing Today's Semi-Structured Gnutella Network. Technical Report CIS-TR-04-02, University of Oregon (2004)
7. Singla, A., Rohrs, C.: Ultrapeers: Another Step Towards Gnutella Scalability. Gnutella Developer's Forum (2002)
8. Lime Wire LLC: Crawler Compatability. Gnutella Developer's Forum (2003)
9. Adamic, L.A., Lukose, R.M., Huberman, B., Puniyani, A.R.: Search in Power-Law Networks. Physical Review E **64** (2001)