Temporal Correlation of Gossiping-based Peer Sampling Methods

Mario Barchetti, Csaba Kiraly
DISI, University of Trento, Italy
mario.barchetti@gmx.com, kiraly@disi.unitn.it

I. INTRODUCTION

Whether used for file distribution, for video streaming, or for other services, every large scale peer-to-peer distributed system should build and maintain a topology. Apart from cases where neighborhood relations are handled by a centralized tracker, peers need information about the existence and routable ID of other peers in order to build their neighborhood.

The most common way of obtaining these peer IDs is through a peer sampling service. The goal of a peer sampling service is to provide random samples of the set of peers present in the whole system at each invocation. The service itself is usually implemented as a distributed system, running independently from the topology management algorithm, exchanging its own messages. Implementations are based on either gossiping [1], [2] or random walk [3] algorithms. In what follows we study the former.

A well known gossiping algorithm is Newscast [4]. While originally developed for information aggregation and dissemination, it can also be used for peer sampling, as shown in [1]. Newscast maintains a fixed size cache at each peer, containing a list of routable peer IDs with associated timestamps. This list can be interpreted as the neighborhood of the peer at the gossiping level, even if it does not necessarily correspond to the peer’s neighborhood at the P2P application level.

Peers periodically select (uniform random) one of the peers from the cache and send a query message, including the content of the cache itself and their own ID with a fresh timestamp. The other peer responds with its own cache and ID, and caches are merged on both sides keeping the freshest entries only.

Another well known algorithm is Cyclon, introduced in [2]. In this case, instead of sending to a random member of the cache, the message is sent to the oldest entry, and instead of merging caches, parts of the cache are being swapped between the two. Several other varieties of gossiping algorithms are explored in [5]. For some of these, authors also evaluate the randomness of generated peer ID samples, finding that if only the oldest entry of the cache is considered, the series can be seen as coming from a uniform random distribution.

Yet, not much is known about the randomness of entries in caches and their temporal correlation. In this work we test the temporal correlation between samples provided by some well known peer sampling methods, through emulation, using a real open source implementation.

This work is funded by the European Union under the KIC EIT ICT Labs Project Smart Ubiquitous Contents (“SmartUC”)

II. PROBLEM STATEMENT

As [1] has shown for the case of Newscast, due to the inherent topology of the gossiping network, new peers returned by the algorithm might not be really “new”, i.e. already known peers have a higher chance of being returned than yet unknown peers. This effect is illustrated in Figure 1, which shows the total number of peers discovered (in a system with \( N = 1000 \) peers) by a given peer, as a function of time (protocol cycles). Two theoretical curves are also shown: the maximum curve, i.e. when all new caches contain only new, yet unseen IDs; and the curve of a hypothetical uniform random sampling of peers.

Observe how Newscast, with a cache size of \( C = 10 \), is unable to explore the whole network (we have verified that the same holds with other small values of \( C \)). Caches of Cyclon contain more fresh IDs, yet it explores the network slower than uniform random sampling. It is clear from the figure that subsequent caches of a peer are not independent of each other.

III. MEASURING TEMPORAL CORRELATION OF PEER SAMPLES

As result of the gossiping algorithm, a node might keep in its cache the same set of peers for most of the time. This possibility is shown in Figure 2, representing the integral over time of the cache content of a single node. This representation of the local view shows that Newscast, when a cache size of 10 is used, keeps its neighborhood very stable; some peers are present in it during almost the whole measurement period, while others are under-represented. As a consequence, large part of the neighborhood graph remains stable throughout the algorithm. The local views with Cyclon, instead, have a lower deviation for any cache sizes.
As expected and depicted in Figure 3, there is no global imbalance, i.e. integrating also over nodes produces an almost uniform random distribution for both of the protocols.

In order to understand better the underlying mechanisms, we evaluate the autocorrelation between the cache of a peer at a random time and every cache at the same peer in subsequent cycles of the protocol. The correlation is interpreted as the ratio of entries contained in both caches to the cache size, and the collected values are averaged over several runs and over all the peers. Note that the average correlation between two uniform random samples would be larger than zero, and its value depends on both $N$ and $C$.

Figure 4 illustrates how much the cache size can influence the temporal correlation of the peer samples, that is the average autocorrelation function of the local view, and highlights again the poor performances of Newscast with a cache size of only 10 nodes. In general, Cyclon performs better (gives lower autocorrelation) than Newscast, but both gossiping protocols seem to perform better in this respect, except for the case when cache size is large, subsequent caches are used in the sampling, but only the last element of the cache is being used.

If only the last element of the cache is considered for peer sampling (as proposed in [5]), autocorrelation converges significantly faster, as seen in Figure 5. Note, however, that in this case the protocol exchanges a whole cache to get only one useful ID.

For larger cache sizes, it is interesting to note that on the one hand, Cyclon can reduce the autocorrelation to its minimum in approx. 7 cycles, while Newscast is much slower. On the other hand, autocorrelation between adjacent caches is lower with Newscast than with Cyclon.

We have shown how subsequent samples seen by a peer are correlated among each other for both protocols and with different cache sizes. These samples are far from being the equivalent of a uniform random sampling.

Although Newscast can be used as a peer sampler, Cyclon seems to perform better in this respect, except for the case when cache size is large, subsequent caches are used in the sampling, but only the last element of the cache is being used.

### IV. Testbed

Every test has been accomplished in an emulated environment on the same host\(^1\), during a period of $30000 \cdot \Delta^2$, and with 1000 clients running a topology test with the GRAPES library [6], implemented in C.

### REFERENCES


---

\(^{1}\)CPU: Xeon E5420 (4 cores at 2.5 GHz). RAM: 8 GB.  
\(^{2}\)\(\Delta\) is the length of the interval between two sent queries.